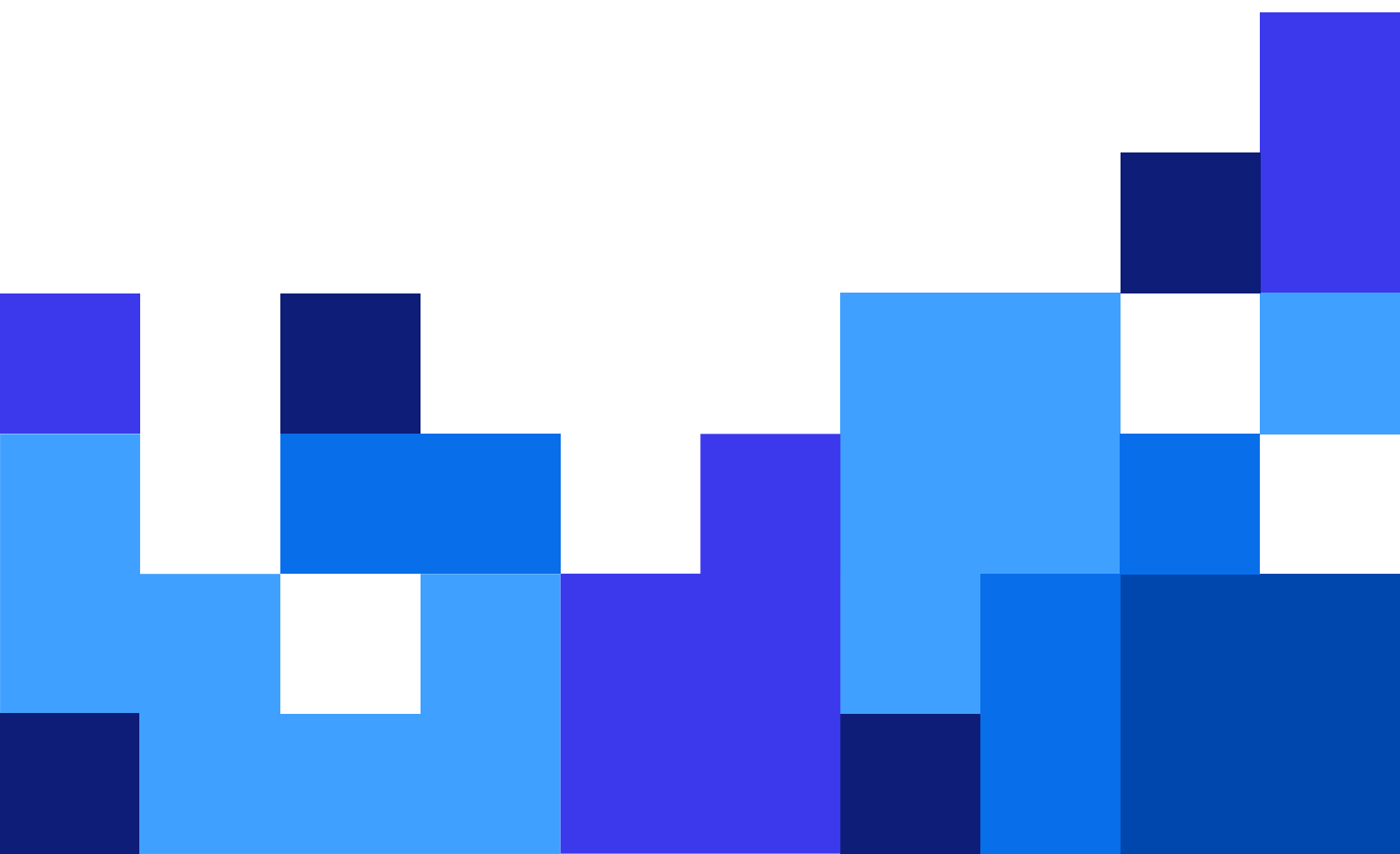


# Guías del usuario de NiceLabel Automation 10

Rev-2024-3



# Tabla de contenidos

<b>1. Bienvenido a NiceLabel Automation</b> .....	<b>7</b>
1.1. Arquitectura .....	8
1.2. Requisitos del sistema .....	9
1.3. Instalación .....	9
1.4. Activación .....	10
1.5. Modo de prueba .....	10
1.6. Pestaña Archivo .....	11
1.6.1. Abrir .....	11
1.6.2. Compatibilidad con productos NiceWatch .....	12
1.6.3. Guardar .....	13
1.6.4. Guardar como .....	14
1.6.5. Opciones .....	14
1.6.6. Acerca de .....	22
<b>2. Comprender los filtros</b> .....	<b>24</b>
2.1. Configurar filtros de texto estructurado .....	26
2.1.1. Filtros de texto estructurado .....	26
2.1.2. Definir campos .....	27
2.1.3. Habilitar estructura dinámica .....	30
2.2. Configurar filtros de datos no estructurados .....	32
2.2.1. Filtros de datos no estructurados .....	32
2.2.2. Definir campos .....	35
2.2.3. Definir subáreas .....	40
2.2.4. Definir áreas de asignación .....	42
2.3. Configurar filtro XML .....	44
2.3.1. Filtro XML .....	44
2.3.2. Definir campos XML .....	45
2.3.3. Definir elementos reiterativos en el filtro XML .....	48
2.3.4. Definir área de asignación XML .....	51
2.4. Configurar filtro JSON .....	54
2.4.1. Filtro JSON .....	54
2.4.2. Definir campos JSON .....	56
2.4.3. Definir elementos reiterativos en el filtro JSON .....	58
2.4.4. Definir área de asignación JSON .....	63
2.5. Configurar nombres de etiqueta e impresora desde datos de entrada .....	66
<b>3. Comprender los desencadenadores</b> .....	<b>68</b>
3.1. Desencadenador de archivo .....	71
3.2. Desencadenador de puerto de serie .....	75
3.3. Desencadenador de base de datos .....	78
3.4. Desencadenador de servidor TCP/IP .....	87
3.5. Desencadenador de cliente TCP/IP .....	92
3.6. Desencadenador de servidor HTTP .....	96
3.7. Desencadenador de servicio Web .....	103
3.8. Desencadenador de programador .....	113

3.8.1. General .....	114
3.8.2. Recurrencias .....	115
3.9. Desencadenador LPS .....	117
3.10. Probar desencadenadores .....	117
3.11. Usar capa de transporte seguro (HTTPS) .....	121
3.12. Proteger configuración de desencadenadores en edición .....	125
3.13. Configuración del firewall para desencadenadores de red .....	125
<b>4. Variables .....</b>	<b>127</b>
4.1. Usar valores compuestos .....	128
4.2. Variables internas .....	129
4.3. Variables globales .....	132
<b>5. Acciones .....</b>	<b>133</b>
5.1. Definir acciones .....	133
5.2. Acciones anidadas .....	133
5.3. Ejecución de acciones .....	134
5.4. Acciones condicionales .....	134
5.5. Identificar acciones en el estado de error de configuración .....	135
5.6. Deshabilitar acciones .....	135
5.7. Copiar acciones .....	136
5.8. Navegar en la lista de acciones .....	136
5.9. Describir las acciones .....	136
5.10. General .....	137
5.10.1. Abrir etiqueta .....	137
5.10.2. Imprimir etiqueta .....	139
5.10.3. Ejecutar archivo de comando XML de Oracle .....	143
5.10.4. Ejecutar todo archivo de comando XML de SAP .....	145
5.10.5. Ejecutar archivo de comando .....	147
5.10.6. Enviar comandos personalizados .....	149
5.11. Impresora .....	151
5.11.1. Establecer impresora .....	151
5.11.2. Establecer nombre de trabajo de impresión .....	152
5.11.3. Redireccionar impresión a archivo .....	154
5.11.4. Establecer parámetro de impresión .....	157
5.11.5. Redireccionar impresión a PDF .....	164
5.11.6. Estado de impresora .....	166
5.11.7. Almacenar etiqueta en impresora .....	171
5.11.8. Imprimir documento PDF .....	173
5.12. Variables .....	175
5.12.1. Establecer variable .....	175
5.12.2. Guardar datos de variable .....	177
5.12.3. Cargar datos de variable .....	179
5.12.4. Manipulación de cadenas .....	181
5.13. Impresión por lotes .....	184
5.13.1. Para bucle .....	184
5.13.2. Usar filtro de datos .....	186
5.13.3. Para cada registro .....	189

5.14. Datos y conectividad .....	192
5.14.1. Abrir documento/programa .....	192
5.14.2. Guardar datos en el archivo .....	194
5.14.3. Leer datos desde el archivo .....	196
5.14.4. Eliminar archivo .....	198
5.14.5. Ejecutar instrucción SQL .....	200
5.14.6. Enviar datos a puerto TCP/IP .....	206
5.14.7. Enviar datos a puerto de serie .....	209
5.14.8. Leer datos desde puerto de serie .....	211
5.14.9. Enviar datos a impresora .....	213
5.14.10. Solicitud de HTTP .....	215
5.14.11. Servicio Web .....	219
5.15. Otros .....	223
5.15.1. Obtener información de etiqueta .....	223
5.15.2. Ejecutar secuencia de comandos .....	228
5.15.2.1. Editor de secuencias de comandos .....	230
5.15.3. Mensaje .....	232
5.15.4. Verificar licencia .....	234
5.15.5. Intentar .....	236
5.15.6. Transformar XML .....	239
5.15.7. Grupo .....	242
5.15.8. Registro de eventos .....	243
5.15.9. Generar vista previa de etiqueta .....	245
5.15.10. Crear variante de etiqueta .....	247
<b>6. Ejecutar y administrar desencadenadores .....</b>	<b>250</b>
6.1. Implementar configuración .....	250
6.2. Opciones de registro de eventos .....	251
6.3. Administrar desencadenadores .....	252
6.4. Usar registro de eventos .....	254
6.5. Si su configuración no se carga... .....	257
<b>7. Opciones de rendimiento y comentarios .....</b>	<b>260</b>
7.1. Procesamiento paralelo .....	260
7.2. Almacenar archivos en caché .....	261
7.3. Tratamiento de errores .....	265
7.4. Modo de impresión sincronizada .....	266
7.4.1. Modo de impresión asincrónica .....	266
7.4.2. Modo de impresión sincronizada .....	267
7.5. Comentarios sobre el estado del trabajo de impresión .....	269
7.6. Exclusión de impresoras de la impresión automática .....	272
7.7. Usar modo de impresión almacenar/recuperar .....	273
7.8. Clúster (conmutación por error) de alta disponibilidad .....	274
7.9. Clúster de equilibrio de carga .....	275
<b>8. Comprender las estructuras de datos .....</b>	<b>277</b>
8.1. Archivos binarios .....	277
8.1.1. Ejemplo .....	277
8.2. Archivos de comandos .....	278

8.2.1. Ejemplo .....	278
8.3. Compuesto CSV .....	279
8.3.1. Ejemplo .....	279
8.4. Datos heredados .....	279
8.4.1. Ejemplo .....	279
8.5. Base de datos de texto .....	280
8.5.1. Ejemplo .....	280
8.6. Datos XML .....	281
8.6.1. Ejemplos .....	281
8.7. Datos JSON .....	283
<b>9. Referencia y resolución de problemas .....</b>	<b>287</b>
9.1. Tipos de archivo de comando .....	287
9.1.1. Especificaciones de archivos de comandos .....	287
9.1.2. Archivo de comando CSV .....	287
9.1.2.1. Ejemplo de Archivo de comando CSV .....	287
9.1.3. Archivo de comando JOB .....	288
9.1.3.1. Ejemplo de Archivo de comando JOB .....	288
9.1.4. Archivo de comando XML .....	289
9.1.4.1. Ejemplo de Archivo de comando XML .....	289
9.1.5. Especificaciones de XML de Oracle .....	294
9.1.5.1. XML DTD .....	294
9.1.5.2. Ejemplo de XML de Oracle .....	295
9.1.6. Todas las especificaciones de SAP XML .....	296
9.1.6.1. Ejemplo de SAP Todo XML .....	296
9.1.7. Archivo de comandos PAS .....	297
9.1.7.1. Ejemplo de archivo de comandos PAS .....	297
9.2. Comandos personalizados .....	297
9.2.1. Usar comandos personalizados .....	297
9.3. Acceso a recursos de red compartidos .....	305
9.4. Almacenamiento de documentos y versiones de archivos de configuración .....	306
9.5. Acceso a bases de datos .....	307
9.5.1. Windows de 32 bits .....	307
9.5.2. Windows de 64 bits .....	307
9.6. Reemplazo de fuente automático .....	308
9.7. Automatización de informes .....	310
9.7.1. Creación de bases de datos temporales .....	311
9.7.2. Diseño de informes automatizados .....	312
9.7.3. Creación de filtros de datos .....	312
9.7.4. Creación de desencadenadores para los nuevos filtros de datos .....	313
9.8. Cambiar configuración predeterminada de impresión multiproceso .....	314
9.9. Compatibilidad con productos NiceWatch .....	315
9.10. Control del servicio Automation con parámetros de línea de comandos .....	317
9.11. Reemplazo de cadena de conexión de bases de datos .....	321
9.12. Introducción de caracteres especiales (Códigos de control) .....	322
9.13. Lista de códigos de control .....	323
9.14. Licencias y uso de impresora .....	324

9.15. Ejecutar en modo de servicio .....	325
9.16. Orden de búsqueda de los archivos solicitados .....	328
9.17. Asegurar el acceso a sus desencadenadores .....	329
9.18. Impresión de sesiones .....	330
9.19. Sugerencias y trucos para usar variables en acciones .....	332
9.20. Modo de rastreo .....	333
9.21. Comprender la configuración de la impresora y DEVMODE .....	335
9.22. Usar la misma cuenta de usuario para configurar y ejecutar desencadenadores .....	336

# 1. Bienvenido a NiceLabel Automation

Automation es una aplicación que automatiza tareas repetitivas.

NiceLabel Automation lo ayuda a implementar y ejecutar un sistema de impresión de etiquetas comerciales óptimo gracias a la sincronización de los eventos comerciales con la producción de etiquetas. La impresión automática sin interacción humana es por mucho el modo más eficaz de evitar errores de los usuarios y maximizar el rendimiento.

La automatización de la impresión de etiquetas usando una aplicación basada en desencadenadores gira en torno a los siguientes tres procesos centrales:

## **Desencadenador**

Los desencadenadores son una función simple pero poderosa que lo ayuda a automatizar el trabajo. En esencia, un desencadenador es una declaración de causa y efecto: "si se produce un evento monitoreado, se debe hacer algo."

Estamos hablando del procesamiento tipo **SI... ENTONCES**. Los desencadenadores sirven para tratar eventos repetitivos.

La impresión automática de etiquetas se desencadena por una operación comercial. NiceLabel Automation está configurado para supervisar una carpeta, archivo o puerto de comunicación. Cuando se realiza una operación comercial, se detecta un cambio de archivo o ingreso de datos. Esto desencadena el proceso de impresión de etiquetas.

Obtenga más información sobre los diversos [Desencadenador](#):

- Desencadenador de archivo
- Desencadenador de puerto de serie
- Desencadenador de base de datos
- Desencadenador de programador
- Desencadenador de servidor TCP/IP
- Desencadenador de cliente TCP/IP
- Desencadenador HTTP
- Desencadenador de servicio Web
- Desencadenador en la nube

## Posición y extracción de datos

Una vez que un desencadenador inicia la impresión, NiceLabel Automation extrae los datos de la etiqueta y los inserta en objetos variables que se colocan en una etiqueta.

Los [Filtros](#) de extracción de datos admiten:

- Archivos de texto estructurado
- Archivos de texto desestructurado
- Varios archivos XML
- Archivos JSON

## Ejecución de acciones

Después de hacer coincidir los datos con objetos variables en una etiqueta, NiceLabel Automation comienza a realizar acciones. El conjunto básico de acciones generalmente incluye **Abrir etiqueta** e **Imprimir etiqueta** para imprimir los datos extraídos en una etiqueta. También puede enviar los datos a destinos personalizados, como archivos en el disco, servidores web, dispositivos de hardware y mucho más.

Hay más de 30 acciones disponibles. Juntas, cubren una amplia mayoría de escenarios comunes en los entornos empresariales de hoy.

Obtenga más información sobre las [Acciones](#) de impresión básicas y avanzadas.

# 1.1. Arquitectura

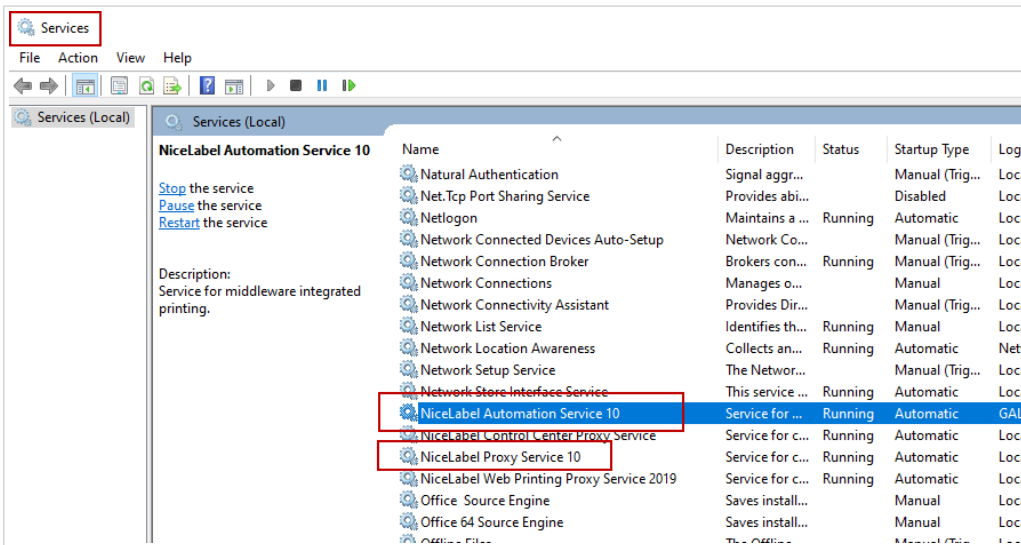
NiceLabel Automation es una aplicación basada en servicio. La ejecución de todas las reglas y acciones se lleva a cabo como proceso de fondo, según las credenciales de la cuenta de usuario definida para el Servicio.

NiceLabel Automation consiste de tres componentes.

- **Automation Builder:** El desarrollador usa esta aplicación para crear desencadenadores, filtros y acciones, y usarlos en una configuración en funcionamiento. Las acciones que pertenecen a dicha configuración se ejecutan después de que un desencadenador recibe los datos. Esta aplicación siempre se ejecuta como la aplicación de 32 bits.
- **Automation Manager:** Esta es la aplicación de administración que monitorea la ejecución del desencadenador en tiempo real y ordena al desencadenador que inicie o se detenga. Automation Manager siempre se ejecuta como aplicación de 32 bits.
- **NiceLabel Automation Servicio:** Es el "motor de impresión" que ejecuta las reglas definidas en los desencadenadores. En realidad, hay dos aplicaciones de servicios: NiceLabel Automation Servicio y NiceLabelServicio de Proxy. El Servicio siempre detecta el valor de bits de la máquina con Windows



y lo ejecuta en el mismo nivel (por ejemplo, como aplicación de 64 bits en Windows de 64 bits), mientras que el Servicio de Proxy siempre lo ejecuta como proceso de 32 bits.



## 1.2. Requisitos del sistema



### NOTA

Es importante revisar los requisitos del sistema más recientes en esta página web: [Requisitos del sistema.](#)

## 1.3. Instalación



### NOTA

A continuación se presenta la versión resumida del procedimiento de instalación. Para obtener más información, consulte el tema NiceLabel Automation [Guía de instalación.](#)

Antes de comenzar con la instalación, asegúrese de que su infraestructura sea compatible con [Requisitos del sistema.](#)

Para instalar NiceLabel Automation, realice lo siguiente:

1. Descargue el archivo de instalación de NiceLabel License Center, luego ejecute el archivo ejecutable.
2. Siga las instrucciones del **Asistente de configuración.**  
Durante la instalación, la Configuración solicitará el nombre de usuario con el cual NiceLabel Automation

NiceLabel Automation se ejecutará el servicio. Asegúrese de seleccionar un nombre de usuario real, ya que el servicio hereda privilegios asociados con el nombre de usuario específico. Para obtener más información, consulte el tema [Ejecutar en modo de servicio](#).

### Actualización de la versión

Para actualizar NiceLabel Automation a la versión más reciente, instale la nueva versión sobre la que está instalada, es decir, sobrescribala. Durante la actualización, la versión anterior será eliminada y reemplazada con la nueva. Se mantendrá la configuración existente. La actualización borra el contenido de la base de datos de registro.

## 1.4. Activación

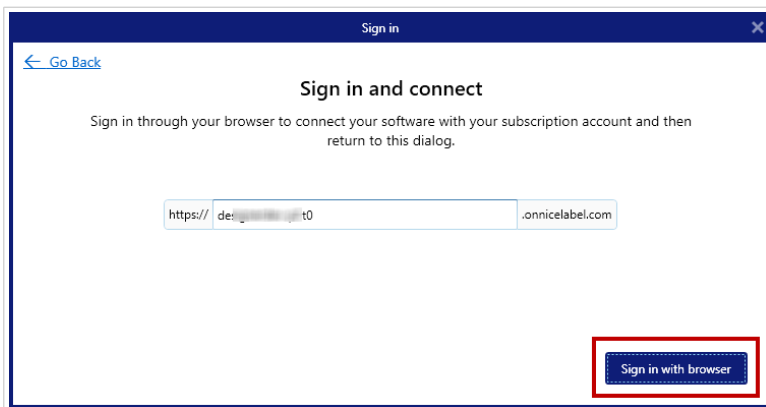
Conectar Automation a License Center para permitir el procesamiento de activadores configurados. El procedimiento de activación requiere una conexión a Internet.



### NOTA

Puede activar el software desde Automation Builder o desde Automation Manager.

cuando abres Automation por primera vez, **Iniciar sesión** Aparece una ventana donde su License Center La dirección ya está ingresada. Hacer clic **Iniciar sesión con el navegador**, luego siga las instrucciones en la pestaña del navegador recién abierta con la solicitud de inicio de sesión.



Para obtener más información sobre cómo activar Automation, ver [el License Center Guía del usuario](#).

## 1.5. Modo de prueba

El modo de prueba permite probar NiceLabel Automation durante hasta 30 días. El modo de prueba ofrece el mismo conjunto de características que la versión con licencia, y por lo tanto permite a evaluación completa del producto antes de adquirirlo. Automation Manager mostrará continuamente el mensaje de aviso de prueba y el número de días de prueba restantes. Una vez que el modo de prueba caduca, el

servicio de NiceLabel Automation deja de procesar desencadenadores. La cuenta regresiva de los 30 días comienza el día de la instalación.



## NOTA

Puede extender el modo de prueba contactándose con su revendedor de NiceLabel y solicitando otra clave de licencia de prueba. Deberá activar la clave de licencia de prueba. Para obtener más información, consulte el tema [Activación](#).

## 1.6. Pestaña Archivo

La pestaña **Archivo** sirve como panel de administración de documentos. Están disponibles las opciones incluidas a continuación:

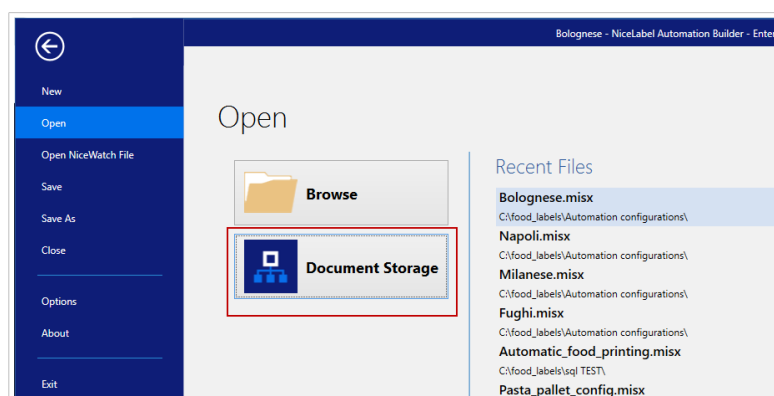
- **Nuevo:** crea un nuevo archivo de configuración.
- **Abrir:** abre archivos de configuración existentes.
- **Abrir archivo NiceWatch:** abre una NiceLabel [configuración de NiceWatch](#) heredada.
- **Guardar:** guarda el archivo de configuración activo.
- **Guardar como:** permite guardar el archivo de configuración activo definiendo su nombre y ubicación.
- **Opciones:** abre el cuadro de diálogo para configurar los valores predeterminados del programa.
- **Acerca de:** proporciona información sobre la versión del software y la licencia.
- **Salir:** cierra la aplicación.

### 1.6.1. Abrir

El diálogo **Abrir** le permite abrir configuraciones existentes en Automation Builder.

**Examinar** permite seleccionar los archivos de configuración en discos de redes locales o conectadas.

**Almacenamiento de documentos** abre la ubicación del almacenamiento de documentos del NiceLabel Control Center conectado. La pestaña **Almacenamiento de documentos** le permite [administrar su copia del archivo de configuración almacenado](#).



El campo **Archivos recientes** enumera los últimos archivos de configuración editados. Haga clic en el archivo para abrirlo.

## 1.6.2. Compatibilidad con productos NiceWatch

NiceLabel Automation le permite cargar las configuraciones de desencadenadores que se crearon usando productos NiceWatch heredados. En la mayoría de los casos, puede ejecutar la configuración de NiceWatch en NiceLabel Automation sin ninguna modificación.

Los productos NiceLabel Automation utilizan un nuevo motor de impresión basado en .NET optimizado para un mejor rendimiento y una menor superficie de memoria. El nuevo motor de impresión no admite cada opción de diseño de etiqueta que está disponible en el diseñador de etiquetas. Cada versión nueva de NiceLabel Automation reduce las diferencias, pero es posible que se encuentre con que algunas funciones no están disponibles.

### Resolver problemas de incompatibilidad

NiceLabel Automation le advertirá si intenta imprimir plantillas de etiquetas existentes que contengan la funcionalidad de diseño, que no está disponible en el nuevo motor de impresión.

Si hay incompatibilidades con las plantillas de etiquetas o archivos de configuración de NiceWatch, Automation será notificado acerca de:

- **Compatibilidad con la configuración del desencadenador:** Al abrir la configuración de NiceWatch (archivo .MIS), NiceLabel Automation lo comprueba en comparación con las funciones compatibles. No todas las funciones de los productos NiceWatch están disponibles en NiceLabel Automation. Si bien algunas no están disponibles, otras están configuradas de manera diferente. Si el archivo MIS contiene funciones no compatibles, las verá en una lista. Automation elimina estas funciones de la configuración.  
En este caso, debe abrir el archivo .MIS en Automation Builder y resolver los problemas de incompatibilidad. Deberá usar las funciones NiceLabel Automation disponibles para recrear los elementos de la configuración eliminada.
- **Compatibilidad con plantillas de etiquetas:** Si sus plantillas de etiquetas existentes contienen funciones de motor de impresión no compatibles, tal lo proporcionado por NiceLabel Automation, verá mensajes de error en el panel **Registro** . Esta información se ve en Automation Builder (al diseñar desencadenadores) o en Automation Manager (al ejecutar los desencadenadores).  
En este caso, debe abrir el archivo de la etiqueta en el diseñador de etiquetas y eliminar las funciones no admitidas de la etiqueta.



### NOTA

Para obtener más información acerca de los problemas de incompatibilidad con NiceWatch y los diseñadores de etiquetas, consulte [NiceLabel Automation Compatibility with NiceLabel Designers V6 and NiceWatch V5](#).

### **Abrir la configuración de NiceWatch para editar**

Abra la configuración de NiceWatch existente (archivo .MIS) en Automation Builder y edítela en Automation Builder. Puede guardar la configuración como archivo .MISX solamente.

Para editar la configuración de NiceWatch, realice lo siguiente:

1. Abra Automation Builder.
2. Vaya a **Archivo > Abrir archivo de NiceWatch**.
3. En el cuadro de diálogo **Abrir** , busque el archivo de configuración de NiceWatch (archivo .MIS).
4. Haga clic en **Aceptar**.
5. Si la configuración contiene funciones no compatibles, estas se muestran en una lista. Automation las elimina de la configuración.

### **Abrir la configuración de NiceWatch para ejecución**

Puede abrir la configuración NiceWatch (archivo .MIS) en Automation Manager sin conversión al formato de archivo NiceLabel Automation (archivo .MISX). Si los desencadenadores de NiceWatch son compatibles con NiceLabel Automation, puede comenzar a usarlos enseguida.

Para editar e implementar la configuración de NiceWatch, realice lo siguiente:

1. Abra Automation Manager.
2. Haga clic en el botón **Agregar**.
3. En el cuadro de diálogo **Abrir**, cambie el tipo de archivo por **Configuración de NiceWatch**.
4. Busque el archivo de configuración de NiceWatch (archivo .MIS).
5. Haga clic en **Aceptar**.
6. Automation Manager mostrará el desencadenador de la configuración seleccionada. Para iniciar el desencadenador, selecciónelo y haga clic en el botón **Iniciar**.



#### **NOTA**

Si hay algún problema de compatibilidad con la configuración de NiceWatch, deberá abrirla en Automation Builder y reconfigurarla.

## **1.6.3. Guardar**

**Guardar** guarda la configuración activa usando el mismo nombre de archivo que se usó para abrirla.



## NOTA

Si se abre una configuración por primera vez, **Guardar** lo lleva directamente al cuadro de diálogo **Guardar como**.

### 1.6.4. Guardar como

**Guardar como** permite guardar el archivo de configuración activo definiendo su nombre y ubicación. Puede guardar los archivos de configuración en su disco local o en el **Almacenamiento de documentos** en su NiceLabel Control Center

El campo **Carpetas recientes** presenta las carpetas que se usaron recientemente para guardar los archivos de configuración.

### 1.6.5. Opciones

Utilice la configuración de este cuadro de diálogo para personalizar la aplicación. Seleccione un grupo de los paneles de la izquierda y configure sus ajustes.

#### Carpetas

Este panel le permite seleccionar las carpetas de almacenamiento predeterminadas para etiquetas, formularios, bases de datos y archivos de imágenes. La ubicación predeterminada de las carpetas es la carpeta Documentos del usuario actual. También están las carpetas predeterminadas donde NiceLabel Automation busca los archivos si proporciona nombres de archivos sin la ruta completa. Para obtener más información sobre el orden de búsqueda de archivos, consulte la sección [Orden de búsqueda de los archivos solicitados](#).

Los cambios relacionados con la carpeta se propagan a NiceLabel Automation Servicio en un minuto. Para aplicar los cambios inmediatamente, puede reiniciar el servicio.



## NOTA

La configuración que aplique aquí se guardará en el perfil del usuario que tenga sesión iniciada. Si el Servicio NiceLabel Automation se ejecuta bajo otra cuenta de usuario, primero tendrá que iniciar sesión en Windows con esa otra cuenta y luego cambiar la carpeta de etiquetas predeterminada. También puede utilizar la utilidad de línea de comandos RUNAS de Windows para ejecutar Automation Builder como ese otro usuario.

## Idioma

El panel Idioma permite seleccionar el idioma de la interfaz de NiceLabel Automation. Seleccione el idioma correspondiente y haga clic en **Aceptar**.



### NOTA

El cambio se aplicará cuando reinicie la aplicación.

## Variables globales

El panel Variables globales define qué ubicación con variables globales almacenadas debe utilizarse:

- **Utilizar las variables globales almacenadas en el servidor (NiceLabel Control Center):** establece la ubicación de almacenamiento de variables globales en NiceLabel Control Center.

<b>EDITION AVAILABILITY:</b>	Automation Easy	Automation Pro	Automation Enterprise
	✗	✓	✓

- **Utilizar las variables globales almacenadas en un archivo (local o compartido):** establece la ubicación de almacenamiento de variables globales en una carpeta local o compartida. Ingrese la ruta de acceso exacta o haga clic en **Abrir** para buscar el archivo.

## Impresoras con licencia



### NOTA

La conservación de la información sobre el uso de impresoras con licencia está disponible con licencias para múltiples usuarios.

El panel **Impresoras con licencia** proporciona información sobre la cantidad de impresoras que se han utilizado en su entorno de impresión.

El grupo **Información sobre impresoras con licencia** muestra cuántos puertos de impresora permitidos están en uso para la impresión en múltiples impresoras.

- **Cantidad de impresoras permitidas por licencia:** cantidad de impresoras que se permiten utilizar con la licencia NiceLabel 10 actual.
- **Cantidad de impresoras utilizadas en los últimos 7 días:** cantidad de impresoras que se han utilizado con NiceLabel 10 durante los últimos 7 días.



### SUGERENCIA

Durante un período de 7 días, la licencia NiceLabel 10 permite utilizar únicamente la cantidad especificada de impresoras distintas.



### AVISO

La licencia comprada define la cantidad de impresoras permitidas. Cuando supera este número, aparece una advertencia. Cuando duplica la cantidad de impresoras permitidas, ya no puede imprimir en impresoras adicionales.

Consulte los estados de impresión en múltiples columnas:

- **Impresora:** nombre o modelo de la impresora seleccionada para el trabajo de impresión.



### NOTA

Si la impresora conectada es compartida, se muestra únicamente el modelo.

- **Ubicación:** nombre de la computadora desde la que se envió el trabajo de impresión.
- **Puerto:** puerto utilizado por la impresora.
- **Último Uso:** tiempo transcurrido desde el último trabajo de impresión.
- **Reservada:** evita que la impresora se elimine después de estar inactiva por más de 7 días.





## NOTA

Si una impresora permanece inactiva por más de 7 días, se eliminará automáticamente a menos que esté habilitada la opción **Reservada**.

El panel **Permisos** le permite bloquear el uso de la impresora en una estación de trabajo local.

- **Esta estación de trabajo solo puede utilizar impresoras reservadas:** con esta opción habilitada, solo las impresoras reservadas pueden editar e imprimir etiquetas en NiceLabel 10.



## SUGERENCIA

Utilice esta opción para no superar la cantidad de puestos de impresoras con licencia disponibles al imprimir en impresoras o aplicaciones de imprimir a archivo no deseadas. Reserve impresoras de etiquetas térmicas o láser exclusivas y limite la impresión solo a ellas. Esto le permite garantizar la impresión continua de etiquetas con una licencia multiusuario.

Esta opción también puede habilitarse usando el archivo `product.config`:

1. Navegue a la carpeta Sistema.  
%PROGRAMDATA%\NiceLabel\NiceLabel 10
2. Haga una copia de seguridad del archivo `product.config`.
3. Abra `product.config` en el editor de texto. El archivo tiene una estructura XML.
4. Agregue las siguientes líneas:

```
<Configuration>
  <Activation>
    <ReservePrinters>Example Printer Name</ReservePrinters>
  </Activation>
  <Common>
    <General>
      <ShowOnlyReservedPrinters>True</ShowOnlyReservedPrinters>
    </General>
  </Common>
</Configuration>
```

5. Guarde el archivo. La Impresora de ejemplo está reservada.

## Centro de control

El panel **Centro de control** le permite habilitar y configurar el monitoreo de eventos y trabajos de impresión. El uso de NiceLabel Control Center permite la ejecución de informes de trabajos de impresión y eventos centralizados, y el almacenamiento centralizado de variables globales.



### NOTA

Esta pestaña queda disponible con licencias LMS Pro o LMS Enterprise .

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	X	✓	✓

El grupo **Dirección** define qué servidor de NiceLabel Control Center debe utilizarse.

- **Dirección del servidor del Centro de control:** URL del servidor de NiceLabel Control Center conectado. Puede seleccionar en la lista de servidores detectados automáticamente en la red, o bien ingresar manualmente una dirección de servidor.

El grupo **Monitoreo de eventos** define qué tipos de eventos debe registrar el NiceLabel Control Center conectado.

- **Eventos de impresión:** registra los eventos relacionados con la impresión desde la estación de trabajo.
- **Eventos de error:** registra todos los errores informados.



### NOTA

De manera predeterminada, los Eventos de impresión y los Eventos de error se registran en NiceLabel Control Center.

- **Actividad de desencadenadores:** registra todos los desencadenadores disparados.
- **Eventos de cambio de estado de desencadenador:** registra los cambios de estado de desencadenador producidos por los desencadenadores disparados.

El grupo **Monitoreo de trabajos de impresión** le permite registrar los trabajos de impresión completados y en curso en NiceLabel Control Center.

- **Habilitar registro de trabajos de impresión en el servidor:** activa el registro de trabajos de impresión.
- **Control de impresión detallado:** habilita el monitoreo de los estados informados por la impresora conectada.



## **NOTA**

Hay dos requisitos para que esta opción esté disponible:

- La impresora debe admitir la comunicación bidireccional.
- NiceLabelEl controlador de la impresora debe utilizarse para la impresión.

## Automatización

Esta configuración define las funciones avanzadas de la aplicación.



### NOTA

Los cambios se aplicarán cuando reinicie la aplicación.

## Comunicación de servicio

- **Puerto de comunicación de servicio:** Automation Manager controla el servicio utilizando el protocolo TCP/IP en el puerto seleccionado. Si el puerto predeterminado no es adecuado para que lo use en la computadora, seleccione otro número de puerto. Tenga cuidado de no seleccionar un número de puerto que ya esté en uso por alguna otra aplicación.

## Registro

- **Borrar entradas de registro todos los días a las:** Define el inicio del proceso de mantenimiento. Los eventos antiguos se borran de la base de datos de registro.
- **Borrar las entradas de registro cuando tengan más de (días):** Especifica la retención de eventos en la base de datos de registro. Todos los eventos que tengan más días que los especificados se borrarán de la base de datos en cada evento de mantenimiento.
- **Mensajes de registro:** Especifica el alcance de registro de eventos. Durante las etapas de desarrollo y prueba, es conveniente habilitar el registro detallado. En este caso, habilite todos los mensajes para mejorar el rastreo de la ejecución del desencadenador. Más tarde, en la etapa de producción, minimice la cantidad de registros y habilite únicamente el registro de errores.

## Rendimiento

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	X	X	✓

**Archivos en caché de almacenamiento de documentos y de recursos compartidos en red:** para mejorar el tiempo hasta la primera etiqueta y el rendimiento general, NiceLabel Automation admite el almacenamiento de archivos en caché. Cuando carga etiquetas, imágenes y datos de bases de datos desde recursos de red compartidos, debe obtener todos los archivos requeridos antes de que pueda comenzar el proceso de impresión.



### SUGERENCIA

Cuando habilita el caché local, se elimina el efecto de latencia de red, ya que los archivos de etiquetas e imágenes se cargan desde el disco local.

Automation utiliza la siguiente carpeta local para copiar en caché los archivos remotos:

%PROGRAMDATA%\NiceLabel\NiceLabel 10\FileCache.

- **Intervalos de actualización (minutos):** Define el intervalo de tiempo en que los archivos en caché se sincronizan con los archivos de la carpeta original. Es el límite de tiempo para que el sistema utilice una versión que puede no ser la última.
- **Eliminar archivos en caché no utilizados después de (días):** define el intervalo de tiempo después del cual se eliminan los archivos en caché.



#### NOTA

El almacenamiento de archivos en caché admite formatos de archivos de etiquetas e imágenes. Después de habilitar el almacenamiento de archivos en caché, reinicie el servicio Automation para que los cambios surtan efecto.

- **La carga previa en caché de carpetas de Almacenamiento de documentos** le permite cargar en caché a nivel local los archivos de Control Center las carpetas del Almacenamiento de documentos en su computadora. Si la carga previa en caché está habilitada, el contenido de la memoria caché local se sincroniza con las carpetas de almacenamiento de documentos seleccionadas.



#### NOTA

En comparación con el [almacenamiento en caché](#), la carga previa en caché reduce el tiempo de impresión de su primera etiqueta impresa.

Agregue cada carpeta de almacenamiento de documentos en una línea separada.

```
/Labels/Folder1
```

```
/Labels/Folder2
```



#### NOTA

Su Automation Builder debe estar conectado al Control Center para habilitar la sincronización fuera de línea de los archivos en caché.

### Soporte de clústeres

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✗	✗	✓

Esta configuración permite el soporte del tipo de clúster (de conmutación por error) de alta disponibilidad en NiceLabel Automation. Seleccione la carpeta que utilizarán ambos nodos en el clúster para compartir información sobre los estados de los desencadenadores en tiempo real.

## Designer

El panel **Designer** le permite configurar el comportamiento de NiceLabel 10 al abrirse.

- **Mostrar cada documento en su propia ventana:** si esta opción está habilitada, los documentos abiertos adicionalmente aparecen en ventanas separadas de NiceLabel 10. Esto se aplica tanto a los documentos recientemente creados como a los existentes.  
Si decide desactivar esta opción, los documentos abiertos adicionalmente aparecerán dentro de la instancia actualmente activa de NiceLabel 10.
- **Origen de configuración de impresora** le permite elegir el origen de la configuración de la impresora.
  - **Use la configuración de impresora del controlador de la impresora:** seleccione esta opción si prefiere imprimir usando la configuración del controlador de impresora. Este comando le permite ajustar la configuración de la impresora antes de imprimir.
  - **Usar configuración de impresora personalizada guardada en etiqueta:** cada etiqueta puede tener su propia configuración de impresora definida y guardada por el usuario. Seleccione esta opción para usar la configuración predeterminada al imprimir.

### 1.6.6. Acerca de

El cuadro de diálogo **Acerca de** brinda información sobre la licencia del producto NiceLabel, habilita la compra y la activación de la licencia (cuando está en modo de prueba), brinda detalles del software y le permite cambiar el nivel del producto NiceLabel 10.

El grupo **Información de la licencia** incluye:

- **Duración del modo de prueba:** información sobre los días restantes para la evaluación del producto. Este segmento ya no será visible después de comprar y activar la licencia del producto.
- **Comprar licencia:** botón lo dirige a la tienda en línea de NiceLabel.
- **Activar licencia:** este botón abre el cuadro de diálogo de activación de la licencia de NiceLabel 10. Consulte la NiceLabel 10 [guía de instalación](#) para obtener detalles sobre el proceso de activación de la licencia. Después de activar la licencia, este botón se llama **Desactivar licencia** – al hacer clic y confirmar la desactivación, su copia de NiceLabel 10 ya no está activada.
- **Cambiar nivel de producto:** abre el cuadro de diálogo de selección del nivel de producto. Cuando está en modo de prueba, puede escoger y evaluar todos los niveles del producto. Con una licencia activada, puede cambiar el nivel del producto únicamente a niveles inferiores.



#### NOTA

Los cambios de nivel del producto tendrán efecto después de reiniciar la aplicación.

- **Actualizar licencia:** abre el cuadro de diálogo de actualización del nivel del producto. Consulte la [NiceLabel 10 guía de instalación](#) para obtener detalles sobre el proceso de actualización de la licencia.

El grupo **Información de software** incluye información sobre la versión del software instalado y el número de compilación.

## 2. Comprender los filtros

NiceLabel Automation utiliza filtros para definir la estructura de los datos recibidos por los desencadenadores. Cada vez que un desencadenador recibe datos, uno o más filtros analizan los datos recibidos. Este proceso extrae valores relevantes para su configuración. Cada filtro incluye reglas para identificar campos dentro de los datos recibidos.



### NOTA

Como resultado, el filtro proporciona una lista de campos y sus valores (pares `nombre:valor`).

### Tipos de filtros

Para obtener más información, consulte las secciones [Filtros de texto estructurado](#), [Filtros de datos no estructurados](#), [Filtro XML](#), y [Filtro JSON](#).

### Estructura de datos

La complejidad del filtro depende de la estructura de los datos. Los archivos con datos estructurados, como archivos CSV o XML, simplifican la extracción de datos. En este caso, los nombres de los campos ya están definidos con los datos. La extracción de los pares `nombre:valor` es rápida. En el caso de los datos sin una estructura clara, lleva más tiempo definir las reglas de extracción. Es posible que se encuentre con estos datos al exportar documentos e informes desde sistemas heredados, interceptar comunicación entre dispositivos y capturar secuencias de impresión.

El filtro define una lista de campos que se extraerán de los datos entrantes una vez que ejecute el filtro.

NiceLabel Automation admite varios tipos de datos de entrada los cuales pueden ser todos analizados por uno de los tipos de filtros compatibles. Asegúrese de escoger el filtro correcto para que coincida con el tipo de datos de entrada. Por ejemplo, podría usar el **filtro de texto estructurado** para los datos de entrada CSV y el **filtro JSON** para los datos de entrada JSON y el **filtro XML** para los datos de entrada XML. Para cualquier dato no estructurado podría utilizar el **filtro de datos no estructurados**. Para obtener más información, consulte el tema [Comprender las estructuras de datos](#).

### Extraer datos

Los filtros son solo un conjunto de reglas y no realizan extracciones por sí mismos. Para ejecutar el filtro, debe ejecutar la acción [Usar filtro de datos](#). La acción ejecutará las reglas del filtro a partir de los datos y extraerá los valores.

Cada desencadenador puede ejecutar tantas acciones Usar filtro de datos como necesite. Si recibe datos de entrada compuestos que un solo filtro no puede analizar, puede definir varios filtros y ejecutar sus reglas en las acciones Usar filtro de datos, ejecutando una tras otra. Al final, puede utilizar los valores extraídos de todas las acciones en la misma etiqueta.



## Asignar campos a variables

Para usar los valores extraídos, debe guardarlos en variables. La acción Usar filtro de datos hace las dos cosas: extrae valores y los guarda en variables. Para configurar este proceso, tiene que signar la variable al campo correspondiente. El valor del campo luego se guardará en una variable asignada.

Field name	Variable name
label_name	LABEL
label_count	QTY
printer_name	PRINTER
art_code	Code
art_name	Article
	Ean
	Weight



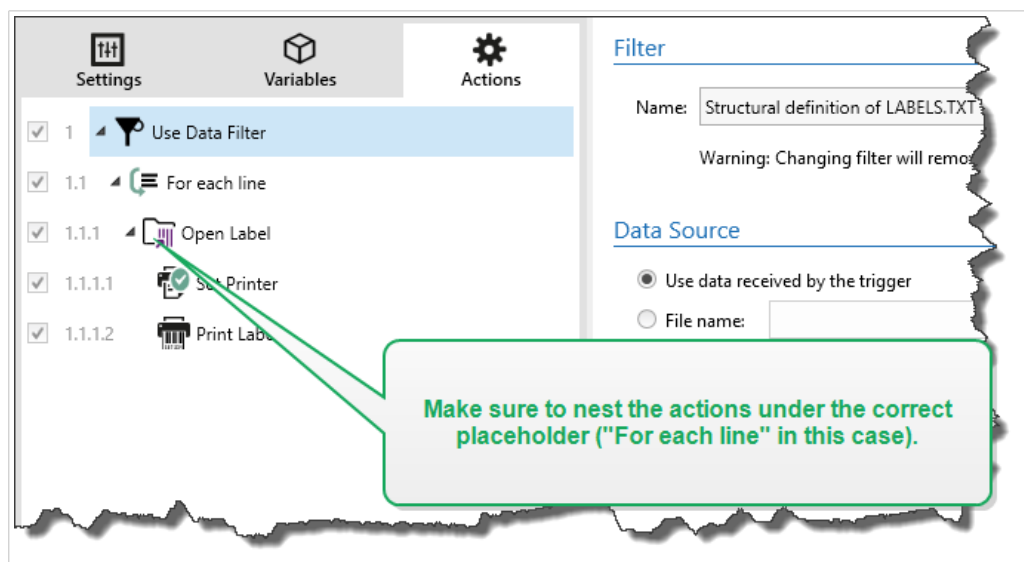
### SUGERENCIA

Es una buena práctica definir campos y variables con los mismos nombres. En este caso, la función de asignación automática vinculará las variables con los campos del mismo nombre, eliminando el proceso manual.

La asignación automática está disponible para todos los tipos de filtros admitidos. Con la asignación automática habilitada, la acción Usar filtro de datos extrae los valores y los asigna automáticamente a las variables con el mismo nombre que los campos. Para obtener más información, consulte la sección [Habilitar estructura dinámica](#) para Filtros de texto estructurado, [Definir área de asignación](#) para Filtros de datos no estructurados y Definir área de asignación para filtros [XML](#) o [JSON](#).

## Ejecutar acciones con datos extraídos

Generalmente, se ejecutan algunas acciones a partir de los datos extraídos, como **Abrir etiqueta**, **Imprimir etiqueta** o algunas de las acciones de conexión de salida. Es muy importante que anide sus acciones debajo de la acción **Usar filtro de datos**. De este modo, se asegura de que las acciones anidadas se ejecuten para cada extracción de datos.



## Ejemplo

Si tiene un archivo CSV con 5 líneas, las acciones anidadas también se ejecutarán 5 veces, una para cada extracción de datos. Si las acciones no se anidan, solo se ejecutarán una vez y tienen ciertos datos de la última extracción de datos. En el ejemplo anterior, la quinta línea del archivo CSV se imprimirá dejando las primeras cuatro líneas sin imprimir. Si utiliza subáreas, asegúrese de anidar su acción debajo del marcador de posición correcto.

## 2.1. Configurar filtros de texto estructurado

### 2.1.1. Filtros de texto estructurado

Para saber más sobre los filtros en general, consulte el tema [Comprender los filtros](#).

Utilice este filtro cuando reciba un archivo de texto estructurado. Son archivos de texto en los que los campos se identifican mediante uno de los métodos:

- **Campos delimitados por un carácter:** Los caracteres delimitadores comunes con la coma o el punto y coma. CSV (valores separados por comas) es un ejemplo típico de un archivo delimitado.
- **Archivos contienen número fijo de caracteres:** En otras palabras, los campos están definidos por columnas de ancho fijo.

Para ver ejemplos de datos de texto estructurado, consulte la sección [Base de datos de texto](#).

## Definir estructuras

Para definir la estructura del archivo de texto, tiene las siguientes opciones:

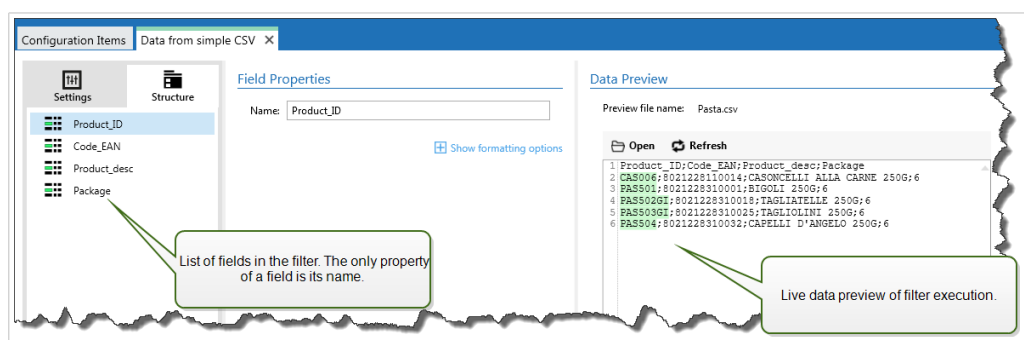
- **Importar estructura usando el Asistente de archivos de texto:** En este caso, haga clic en el botón **Importar estructura de datos** en la banda de opciones y siga las instrucciones en pantalla. Una vez que termina el asistente, se definirán el tipo de base de datos de texto y todos los campos. Si la primera línea de datos tiene los nombres de los campos, el Asistente puede importarlos. Este es el método recomendado, si el desencadenador siempre recibe datos cuya estructura no se modifica.
- **Definir los campos manualmente:** En este caso, debe definir en forma manual el tipo de datos (campos delimitados o campos de ancho fijo) y luego definir los nombres de los campos. Para obtener más información, consulte el tema [Definir campos](#).
- **Leer los campos en forma dinámica:** En este caso, el desencadenador podría recibir datos estructurados de una forma diferente. Un ejemplo de esto son los nombres de campos - la estructura dinámica elimina la necesidad de actualizar el filtro para cada cambio estructural. El soporte de la estructura dinámica lee automáticamente todos los campos de datos, sin importar si existen nuevos campos o si faltan algunos de los campos anteriores. Los asigna automáticamente a las variables usando los mismos nombres. Para obtener más información, consulte el tema [Habilitar estructura dinámica](#).

La sección Vista previa de los datos simplifica la configuración. En el panel de vista previa, el resultado de la regla del filtro definido se resalta en el área de vista previa con cada cambio de configuración. La Vista previa de datos le permite comprobar que datos se extraen con cada regla.

### 2.1.2. Definir campos

Para archivos de texto estructurados, la definición de los campos es muy sencilla. Existen dos opciones:

- **El delimitador define los campos:** en este caso, tiene un carácter delimitador, como coma o punto y coma, que separa los campos. Solo debe definir los nombres de los campos en el mismo orden con el que aparecerán en los datos recibidos por un desencadenador.
- **Campos con ancho fijo:** en este caso, debe definir los nombres de los campos en el mismo orden con el que aparecerán en los datos recibidos por un desencadenador y definir el número de caracteres que ocupará el campo. Esos caracteres serán leídos desde los datos para este campo.



### **Vista previa de los datos**

Esta sección proporciona la vista previa de la definición del campo. Cuando se selecciona el elemento definido, se resalta su posición en los datos en vista previa.

- **Vista previa del nombre del archivo:** especifica el archivo que contiene los datos de muestra que se analizarán a través del filtro. Se copia el archivo en vista previa de la definición del filtro. Si cambia el nombre del archivo en vista previa, se guarda el nuevo nombre.
- **Abrir:** selecciona otro archivo sobre el cual ejecutar las reglas de filtro.
- **Actualizar:** vuelve a ejecutar las reglas de filtro sobre el contenido del nombre del archivo en vista previa. Automation actualiza la sección Vista previa con el resultado.

## Opciones de formato

Esta sección define las funciones de manipulación de cadenas que se aplicarán a las variables o campos seleccionados. Puede seleccionar una o varias funciones. Las funciones se aplican en el mismo orden seleccionado en la interfaz del usuario: de arriba a abajo.

- **Borrar espacios en el comienzo:** borra todos los caracteres de espacio (código 32 ASCII decimal) en el comienzo de la cadena.
- **Borrar espacios en el final:** borra todos los caracteres de espacio (valor 32 ASCII decimal) en el final de una cadena.
- **Borrar el carácter de apertura y cierre:** borra los caracteres de apertura y cierre seleccionados que aparecen por primera vez en la cadena.

### Ejemplo

Si utiliza "{" para el carácter de apertura y "}" para el carácter de cierre, la cadena de entrada `{{selección}}` se convertirá en `{selección}`.

- **Buscar y reemplazar:** ejecuta la función estándar de buscar y reemplazar a partir de los valores provistos para *buscar qué* y *reemplazar con*. También puede usar expresiones comunes.



#### NOTA

Existen varias implementaciones de expresiones comunes en uso. utiliza la sintaxis de .NET Framework para las expresiones comunes. Para obtener más información, consulte [Artículo de base de conocimiento](#).

- **Reemplazar caracteres que no se pueden imprimir con espacios:** reemplaza todos los caracteres de control en la cadena con espacios (código 32 ASCII decimal). Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Borrar caracteres que no se pueden imprimir:** borra todos los caracteres de control de la cadena. Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Descodificar caracteres especiales:** descodifica los caracteres (o códigos de control) que no están disponibles en el teclado, como Retorno de carro y Avance de línea. utiliza una notación para codificar dichos caracteres en lenguaje natural, como `<CR>` para Retorno de carro y `<LF>` para Avance de línea. Para obtener más información, consulte el tema [Introducción de caracteres especiales \(Códigos de control\)](#).

Esta opción convierte caracteres especiales de la sintaxis en caracteres binarios reales.

### Ejemplo

Cuando reciba los datos "`<CR><LF>`", los usa como una cadena simple de 8 caracteres. Deberá habilitar esta opción para interpretar y usar los datos recibidos como dos caracteres binarios `CR` (Retorno de carro – ASCII código 13) y `LF` (Avance de línea – ASCII código 10).

- **Buscar y eliminar todo antes de:** Busca la cadena provista y borra todos los caracteres desde el comienzo de los datos hasta la cadena. La cadena buscada también se puede borrar.
- **Buscar y eliminar todo después de:** Busca la cadena provista y borra todos los caracteres desde la cadena hasta el fin de los datos. La cadena buscada también se puede borrar.
- **Cambiar grafía:** Cambia todos los caracteres de las cadenas a mayúsculas o minúsculas.

### 2.1.3. Habilitar estructura dinámica

El filtro de texto estructurado tiene la capacidad de identificar automáticamente los campos y sus valores dentro de los datos recibidos. Esto elimina la necesidad de asignación manual *variable-a-campo*.

La funcionalidad **Estructura dinámica** es útil si el desencadenador recibe datos con una estructura cambiante. En dichos casos, la estructura principal de los datos permanece sin cambios ( p.ej. los campos están delimitados por coma), o conserva la misma estructura, pero cambia **el orden** o **el número** de los campos. Podría haber campos nuevos, o algunos de los campos anteriores podrían ya no estar disponibles. Debido a la **Estructura dinámica**, habilitada, el filtro identifica automáticamente la estructura del archivo recibido. Al mismo tiempo, el filtro lee los nombres de campos y sus valores (pares `name:value`) de los datos. Esto elimina la necesidad de asignación manual de campos a variables.

[Usar filtro de datos](#) la acción no ofrece posibilidades de asignación, debido a que realiza la asignación en forma dinámica. Incluso no tendrá que definir las variables de etiquetas en la configuración de los desencadenadores. La acción asignará valores de campo a las variables de las etiquetas del mismo nombre, sin tener que importar las variables de la etiqueta. No obstante, esta regla se aplica solamente a la acción [Imprimir etiqueta](#). Si quiere usar los valores de campo en otra acción, deberá definir variables en el desencadenador, y mantener, al mismo tiempo, la asignación automática de *variable a campo*.

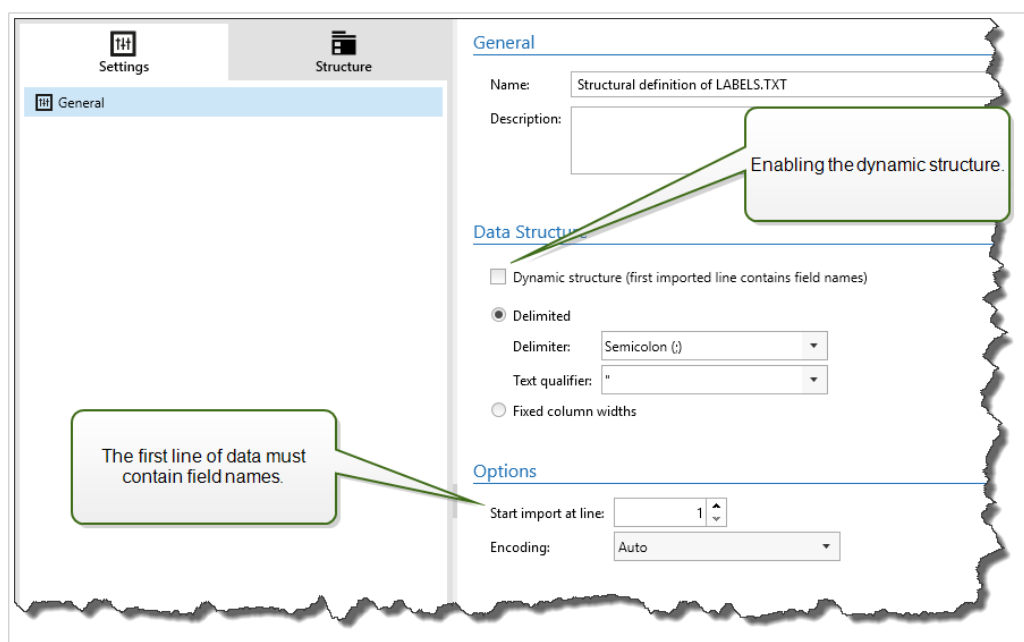


#### NOTA

No se producirá ningún error si el campo disponible en los datos de entrada no tiene una variable de etiqueta correspondiente. ignora en silencio las variables faltantes.

## Configurar la estructura dinámica

Para configurar la estructura dinámica, habilite la opción **Estructura dinámica** en las propiedades del filtro de texto estructurado.



- La primera línea de los datos debe tener los nombres de los campos.
- La línea que seleccione para **Comenzar importación en la línea** debe ser la línea con los nombres de los campos (generalmente, la primera línea en los datos).
- La estructura de los datos debe delimitarse.
- Si es necesario, puede aplicar formato a los datos.

## Opciones de formato

Esta sección define las funciones de manipulación de cadenas que se aplicarán a las variables o campos seleccionados. Puede seleccionar una o varias funciones. Las funciones se aplican en el mismo orden seleccionado en la interfaz del usuario: de arriba a abajo.

- **Borrar espacios en el comienzo:** borra todos los caracteres de espacio (código 32 ASCII decimal) en el comienzo de la cadena.
- **Borrar espacios en el final:** borra todos los caracteres de espacio (valor 32 ASCII decimal) en el final de una cadena.
- **Borrar el carácter de apertura y cierre:** borra los caracteres de apertura y cierre seleccionados que aparecen por primera vez en la cadena.

## Ejemplo

Si utiliza "{" para el carácter de apertura y "}" para el carácter de cierre, la cadena de entrada `{{selección}}` se convertirá en `{selección}`.

- **Buscar y reemplazar:** ejecuta la función estándar de buscar y reemplazar a partir de los valores provistos para *buscar qué* y *reemplazar con*. También puede usar expresiones comunes.



#### NOTA

Existen varias implementaciones de expresiones comunes en uso. utiliza la sintaxis de .NET Framework para las expresiones comunes. Para obtener más información, consulte [Artículo de base de conocimiento](#).

- **Reemplazar caracteres que no se pueden imprimir con espacios:** reemplaza todos los caracteres de control en la cadena con espacios (código 32 ASCII decimal). Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Borrar caracteres que no se pueden imprimir:** borra todos los caracteres de control de la cadena. Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Descodificar caracteres especiales:** descodifica los caracteres (o códigos de control) que no están disponibles en el teclado, como Retorno de carro y Avance de línea. utiliza una notación para codificar dichos caracteres en lenguaje natural, como <CR> para Retorno de carro y <LF> para Avance de línea. Para obtener más información, consulte el tema [Introducción de caracteres especiales \(Códigos de control\)](#).

Esta opción convierte caracteres especiales de la sintaxis en caracteres binarios reales.

#### Ejemplo

Cuando reciba los datos "<CR><LF>", los usa como una cadena simple de 8 caracteres. Deberá habilitar esta opción para interpretar y usar los datos recibidos como dos caracteres binarios CR (Retorno de carro – ASCII código 13) y LF (Avance de línea – ASCII código 10).

- **Buscar y eliminar todo antes de:** Busca la cadena provista y borra todos los caracteres desde el comienzo de los datos hasta la cadena. La cadena buscada también se puede borrar.
- **Buscar y eliminar todo después de:** Busca la cadena provista y borra todos los caracteres desde la cadena hasta el fin de los datos. La cadena buscada también se puede borrar.
- **Cambiar grafía:** Cambia todos los caracteres de las cadenas a mayúsculas o minúsculas.

## 2.2. Configurar filtros de datos no estructurados

### 2.2.1. Filtros de datos no estructurados

Para saber más sobre los filtros en general, consulte el tema [Comprender los filtros](#).



Use este filtro siempre que el desencadenador reciba datos no estructurados, como documentos e informes exportados del sistema heredado, comunicación interceptada entre dispositivos y secuencia de impresión capturada. El filtro le permite extraer campos individuales, campos en las subáreas reiterativas, e incluso pares `nombre-valor`.

Para ver ejemplos de datos de texto estructurado, consulte las secciones [Datos heredados](#), [Compuesto CSV](#) y [Archivos binarios](#).

The screenshot shows the 'Configuration Items' window with the 'H: Header Data' tab selected. The 'Field Properties' section is configured for the field 'DOCNUM'. The 'Field Start' section is set to 'Position in document' with 'Line' at 1 and 'Character' at 11. The 'Field End' section is set to 'Length' with 'Lines' at 0 and 'Characters' at 10. A callout box points to the 'Field End' section with the text 'Extraction logic for each field.' Another callout box points to the list of fields on the left with the text 'List of fields defined as output from the filter.' The 'Data Preview' section shows a preview file named 'CA\Projects\SAP\Certification\...\Example\_FI\_1.rdi'. The preview content is as follows:

```
1 H046A018500000003742D2140_ACC_STAT_01PRINTER WE
2 S
3 CODEPAGE 1100 LANGUAGE DE
4 CPAGEPAGE FIRST
5 DMAIN XKS11
6 DMAIN S11 RF140-
7 DMAIN XS23 ULINE
8 DMAIN S23
9 DMAIN S23
10 DMAIN S23
11 DMAIN S23
12 DMAIN S23
13 DMAIN S23
14 DMAIN
15 DMAIN
16 DMAIN
17 DMAIN
18 DMAIN
19 DMAIN
20 DMAIN
21 DMAIN
22 DMAIN S55
23 DMAIN S55 RF140-
24 DMAIN S55 RF140-
25 DMAIN S55
26 CINC-BEGIN ADRS_SENDER TEXT ADRS DE
27 DADDRESS X
28 DADDRESS
29 DADDRESS ULINE
30 DADDRESS
31 CINC-END ADRS_SENDER TEXT ADRS DE
32 DADDRESS
33 DADDRESS STXADE
34 DADDRESS
```

A callout box points to the preview content with the text 'Highlighting the result of the filter rules.' The preview shows that lines 22, 23, 24, and 25 are highlighted in the original image.

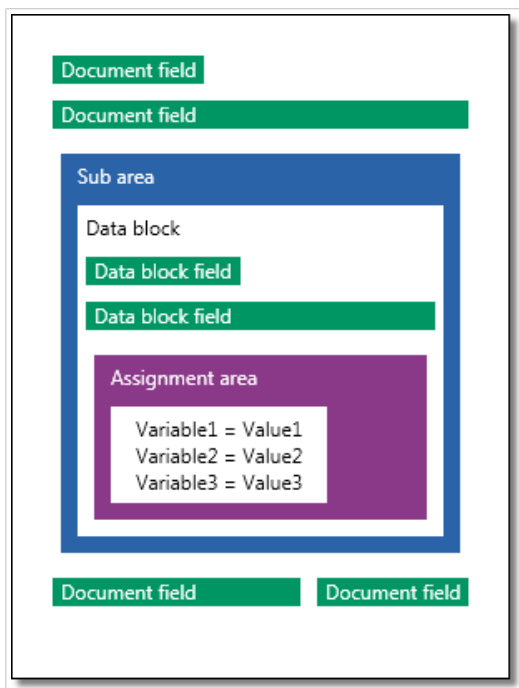
## Definir estructuras

Elementos que puede usar para configurar el filtro:

- **Campo:** Especifica la ubicación de los datos del campo entre la ubicación del campo inicial y final. Hay varias opciones para definir la ubicación del campo, de la codificación rígida de la posición a la habilitación de la posición relativa. Debe asignar los campos definidos a las variables respectivas en la acción [Usar filtro de datos](#). Para obtener más información, consulte el tema [Definir campos](#).
- **Subárea:** Especifica la ubicación de datos reiterativos. Cada subárea define al menos un bloque de datos, que a su vez tiene datos para las etiquetas. Puede haber subáreas definidas dentro de las subáreas, lo que permite la definición de estructuras complejas. Puede definir campos dentro de cada bloque de datos. Debe asignar los campos definidos a las variables respectivas en la acción. Para cada subárea, Automation define un nuevo nivel del marcador de posición dentro de la acción Usar filtro de datos, para que pueda asignar variables a los campos de ese nivel. Para obtener más información, consulte el tema [Definir subáreas](#).
- **Área de asignación:** Especifica la ubicación de datos reiterativos que tienen los pares `nombre-valor`. Automation lee los nombres de campo y sus valores simultáneamente. Automation también realiza automáticamente la asignación a variables. Use esta función para ajustar el filtro a los datos de entrada cambiantes, lo que elimina el tiempo de mantenimiento. Puede definir el área de asignación en el nivel de raíz del documento, o dentro de la subárea. Para obtener más información, consulte el tema [Definir áreas de asignación](#).

La sección Vista previa de los datos simplifica la configuración. El resultado de la regla del filtro definido se resalta en el área de vista previa con cada cambio de configuración. Puede ver qué datos se extraerían con cada regla.

Los campos se pueden definir en el nivel de raíz como campos de documento. Los campos se pueden definir dentro del bloque de datos. Los pares `nombre-valor` se pueden definir dentro del área de asignación.



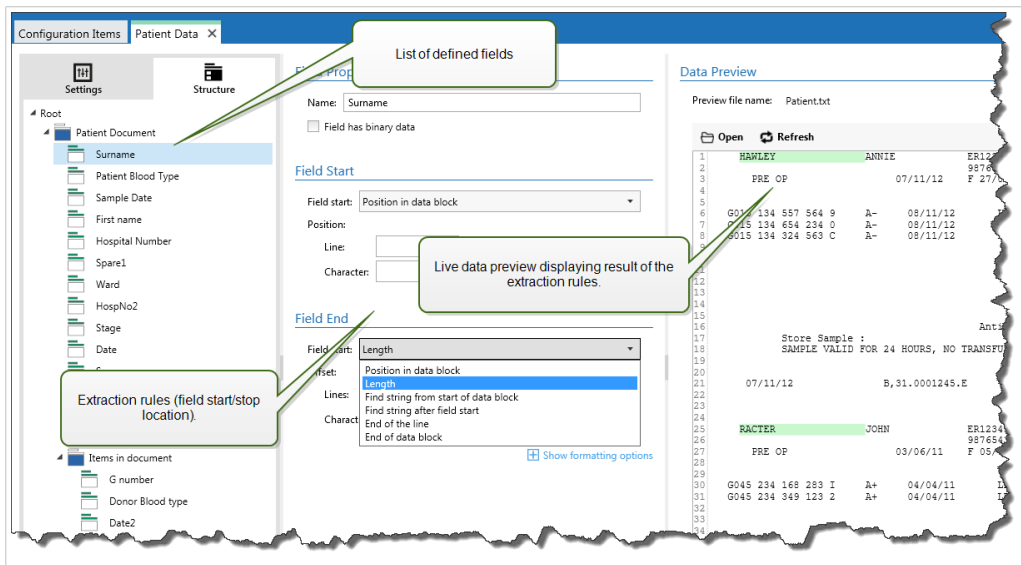
## General

Esta sección define las propiedades generales del filtro de datos no estructurados.

- **Nombre:** Especifica el nombre del filtro. Use un nombre descriptivo que identifique el rol del filtro en una configuración. Puede cambiarlo en cualquier momento.
- **Descripción:** Le permite describir el propósito de este filtro. Puede usarla para escribir una explicación breve de lo que hace el filtro.
- **Codificación:** Especifica la codificación de los datos con los que trabajará este filtro.
- **Ignorar líneas vacías en bloques de datos:** Especifica que no aparece un error si el filtro extrae valores de campos vacíos de los bloques de datos.

### 2.2.2. Definir campos

Cuando define un campo, debe definir su nombre y una regla de cómo extraer los valores de campo de los datos. Cuando se ejecuta el filtro, las reglas de extracción se aplican a los datos de entrada y se asigna el resultado al campo.



## Propiedades del campo

- **Nombre:** Especifica el nombre único del campo.
- **El campo tiene datos binarios:** Especifica que el campo contiene datos binarios. No lo habilite a menos que realmente reciba datos binarios.

## Inicio del campo

- **Posición en documento:** La posición no modificable en los datos determina el punto inicial/final. El origen de coordenada está en la esquina izquierda superior. El carácter en la posición definida está incluido en los datos extraídos.
- **Fin del documento:** El punto inicial/final está al final del documento. También puede definir un desplazamiento desde el final para un número específico de líneas o caracteres.
- **Buscar cadena desde el inicio del documento:** La posición de la cadena buscada define el punto inicial/final. Después de que Automation encuentra la cadena requerida, el siguiente carácter determina el punto inicial/final. La cadena buscada no está incluida en los datos extraídos. La búsqueda predeterminada distingue mayúsculas de minúsculas.
  - **Iniciar búsqueda desde posición absoluta:** Puede ajustar la búsqueda cambiando la posición inicial desde el punto inicial (posición 1,1) para un desplazamiento. Utilice esta función para omitir la búsqueda al comienzo de los datos.
  - **Repetición:** Especifica qué repetición de la cadena buscada debe coincidir. Utilice esta opción si no espera configurar la posición inicial/final después de la primera cadena encontrada.
  - **Desplazamiento desde cadena:** Especifica el desplazamiento positivo o negativo después de la cadena buscada.

### Ejemplo

Debe definir el desplazamiento que incluirá la cadena buscada en los datos extraídos.

## Final del campo

- **Posición en documento:** La posición no modificable en los datos determina el punto inicial/final. El origen de coordenada está en la esquina izquierda superior. El carácter en la posición definida está incluido en los datos extraídos.
- **Fin del documento:** El punto inicial/final está al final del documento. También puede definir un desplazamiento desde el final para un número específico de líneas o caracteres.
- **Buscar cadena desde el inicio del documento:** La posición de la cadena buscada define el punto inicial/final. Después de que Automation encuentra la cadena requerida, el siguiente carácter determina el punto inicial/final. La cadena buscada no está incluida en los datos extraídos. La búsqueda predeterminada distingue mayúsculas de minúsculas.
  - **Iniciar búsqueda desde posición absoluta:** Puede ajustar la búsqueda cambiando la posición inicial desde el punto inicial (posición 1,1) para un desplazamiento. Utilice esta función para omitir la búsqueda al comienzo de los datos.
  - **Repetición:** Especifica qué repetición de la cadena buscada debe coincidir. Utilice esta opción si no espera configurar la posición inicial/final después de la primera cadena encontrada.
  - **Desplazamiento desde cadena:** Especifica el desplazamiento positivo o negativo después de la cadena buscada.

### Ejemplo

Debe definir el desplazamiento que incluirá la cadena buscada en los datos extraídos.

- **Buscar cadena después del inicio del campo:** El punto inicial/final se define según la posición de la cadena buscada, como en la opción **Buscar cadena desde el inicio del documento**, pero la búsqueda comienza después de la posición inicial del campo/área, no en el comienzo de los datos.
- **Longitud:** Especifica la longitud de los datos en las líneas en caracteres. El número especificado de líneas o caracteres se extraerá desde la posición inicial.
- **Fin de la línea:** Especifica si se extraerán los datos desde la posición inicial hasta el final de la misma línea. Puede definir un desplazamiento negativo desde el final de la línea.

## Opciones de formato

Esta sección define las funciones de manipulación de cadenas que se aplicarán a las variables o campos seleccionados. Puede seleccionar una o varias funciones. Las funciones se aplican en el mismo orden seleccionado en la interfaz del usuario: de arriba a abajo.

- **Borrar espacios en el comienzo:** borra todos los caracteres de espacio (código 32 ASCII decimal) en el comienzo de la cadena.
- **Borrar espacios en el final:** borra todos los caracteres de espacio (valor 32 ASCII decimal) en el final de una cadena.
- **Borrar el carácter de apertura y cierre:** borra los caracteres de apertura y cierre seleccionados que aparecen por primera vez en la cadena.

### Ejemplo

Si utiliza "{" para el carácter de apertura y "}" para el carácter de cierre, la cadena de entrada `{{selección}}` se convertirá en `{selección}`.

- **Buscar y reemplazar:** ejecuta la función estándar de buscar y reemplazar a partir de los valores provistos para *buscar qué* y *reemplazar con*. También puede usar expresiones comunes.



#### NOTA

Existen varias implementaciones de expresiones comunes en uso. utiliza la sintaxis de .NET Framework para las expresiones comunes. Para obtener más información, consulte [Artículo de base de conocimiento](#).

- **Reemplazar caracteres que no se pueden imprimir con espacios:** reemplaza todos los caracteres de control en la cadena con espacios (código 32 ASCII decimal). Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Borrar caracteres que no se pueden imprimir:** borra todos los caracteres de control de la cadena. Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Descodificar caracteres especiales:** descodifica los caracteres (o códigos de control) que no están disponibles en el teclado, como Retorno de carro y Avance de línea. utiliza una notación para codificar dichos caracteres en lenguaje natural, como `<CR>` para Retorno de carro y `<LF>` para Avance de línea. Para obtener más información, consulte el tema [Introducción de caracteres especiales \(Códigos de control\)](#).

Esta opción convierte caracteres especiales de la sintaxis en caracteres binarios reales.

### Ejemplo

Cuando reciba los datos "`<CR><LF>`", los usa como una cadena simple de 8 caracteres. Deberá habilitar esta opción para interpretar y usar los datos recibidos como dos caracteres binarios `CR` (Retorno de carro – ASCII código 13) y `LF` (Avance de línea – ASCII código 10).

- **Buscar y eliminar todo antes de:** Busca la cadena provista y borra todos los caracteres desde el comienzo de los datos hasta la cadena. La cadena buscada también se puede borrar.
- **Buscar y eliminar todo después de:** Busca la cadena provista y borra todos los caracteres desde la cadena hasta el fin de los datos. La cadena buscada también se puede borrar.
- **Cambiar grafía:** Cambia todos los caracteres de las cadenas a mayúsculas o minúsculas.

### 2.2.3. Definir subáreas

La subárea es la sección de datos dentro de la cual hay varios bloques de datos identificados por la misma regla de extracción. Cada bloque de datos proporciona datos para una sola etiqueta.

Todos los bloques de datos deben identificarse con la misma regla de configuración. Cada bloque de datos puede tener otra subárea. Puede definir un número ilimitado de subáreas anidadas dentro de subáreas principales.

Cuando el filtro contiene la definición de una subárea, la acción [Usar filtro de datos](#) mostrará subáreas con marcadores de posición anidados. Toda acción anidada debajo de dicho marcador de posición se ejecutará únicamente para los bloques de datos en este nivel. Puede imprimir diferentes etiquetas con datos desde diferentes subáreas.

The screenshot shows a software interface with a 'Configuration Items' pane on the left and a 'Data Preview' pane on the right. The 'Patient Document' is selected in the configuration tree. Annotations with callouts point to specific parts of the interface:

- A callout points to the 'Patient Document' in the configuration tree, stating: "Patient Document is a sub-area containing data for each patient."
- A callout points to the 'Field Start' configuration (Position in data block), stating: "Data block for 1st record (patient 1)".
- A callout points to the 'Field End' configuration (Length), stating: "Data block for 2nd record (patient 2)".

The 'Data Preview' pane shows a text file named 'Patient.txt' with the following content:

```

1 HAWLEY ANNIE ER12345678 AB
2 9876543210
3 PRE OP 07/11/12 F 27/06/47
4
5
6 G015 134 557 564 9 A- 09/11/12 LDBS
7 G015 134 654 234 0 A- 09/11/12 LDBS
8 G015 134 324 563 C A- 09/11/12 LDBS
9
10
11
12
13
14
15
16 Antibody S
17 Store Sample :
18 SAMPLE VALID FOR 24 HOURS, NO TRANSFUSION HIS
19
20 07/11/12 B,31.0001245.E
21
22
23
24
25 RACTER JOHN ER12345679
26 9876543210
27 PRE OP 09/06/11 F 09/02/81
28
29
30 G015 234 168 289 1 A+ 04/04/11 LDBS
31 G015 234 349 123 2 A+ 04/04/11 LDBS
  
```



## Configurar subárea

La subárea se define con reglas similares como campos individuales. Cada subárea se define con los siguientes parámetros.

- **Nombre de subárea:** Especifica el nombre de la subárea.
- **Bloques de datos:** Especifica cómo identificar los bloques de datos dentro de la subárea. Cada subárea tiene al menos un bloque de datos. Cada bloque de datos proporciona datos para una sola etiqueta.
  - **Cada bloque contiene un número fijo de líneas:** Especifica que cada bloque de datos en una subárea tiene el número fijo de líneas provisto. Use esta opción si sabe que cada bloque de datos tiene exactamente el mismo número de líneas.
  - **Bloques que comienzan con una cadena:** Especifica que los bloques de datos comienzan con la cadena provista. Todo contenido entre dos cadenas provistas pertenece a un bloque de datos separado. El contenido entre la última cadena y el final de los datos identifica el último bloque de datos.
  - **Bloques que finalizan con una cadena:** Especifica que los bloques de datos finalizan con la cadena provista. Todo contenido entre dos cadenas provistas pertenece a un bloque de datos separado. El contenido entre el comienzo de los datos y la primera cadena identifica el primer bloque de datos.
  - **Bloques separados por una cadena:** Especifica que los bloques de datos están separados por la cadena provista. Todo contenido entre dos cadenas seleccionadas pertenece a un bloque de datos separado.
- **Comienzo del primer bloque de datos:** Especifica la posición inicial del primer bloque de datos. Al mismo tiempo, define la posición inicial de la subárea. Generalmente, la posición inicial también es el comienzo de los datos recibidos. Los parámetros de configuración son los mismos que para definir los campos. Para obtener más información, consulte el tema [Definir campos](#).
- **Fin del último bloque de datos:** Especifica la posición final del último bloque de datos. Al mismo tiempo, define la posición final de la subárea. Generalmente, la posición final es el final de los datos recibidos. Los parámetros de configuración son los mismos que para definir los campos. Para obtener más información, consulte el tema [Definir campos](#).

## Configurar campos dentro de la subárea

Los campos dentro de la subárea están configurados usando los mismos parámetros que para los campos definidos en el nivel de raíz. Para obtener más información, consulte el tema [Definir campos](#).



### NOTA

Los números de líneas del campo hacen referencia a la posición dentro del bloque de datos, no la posición dentro de los datos de entrada.

### Vista previa de los datos

Esta sección proporciona la vista previa de la definición del campo. Cuando se selecciona el elemento definido, se resalta su posición en los datos en vista previa.

- **Vista previa del nombre del archivo:** especifica el archivo que contiene los datos de muestra que se analizarán a través del filtro. Se copia el archivo en vista previa de la definición del filtro. Si cambia el nombre del archivo en vista previa, se guarda el nuevo nombre.
- **Abrir:** selecciona otro archivo sobre el cual ejecutar las reglas de filtro.
- **Actualizar:** vuelve a ejecutar las reglas de filtro sobre el contenido del nombre del archivo en vista previa. Automation actualiza la sección Vista previa con el resultado.

## 2.2.4. Definir áreas de asignación

El filtro de datos no estructurados identifica automáticamente los campos y sus valores en los datos recibidos. Esto hace que la asignación manual *variable a campo* sea innecesaria.

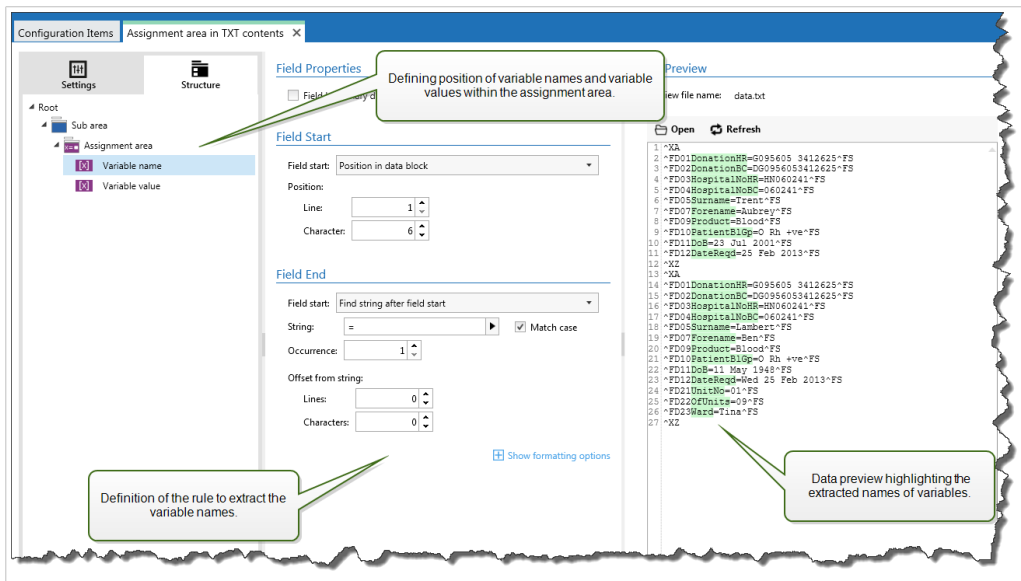
La funcionalidad **Estructura dinámica** es útil si el desencadenador recibe datos con una estructura cambiante. En dichos casos, la estructura principal de los datos permanece sin cambios ( p.ej. los campos están delimitados por coma), o conserva la misma estructura, pero cambia **el orden o el número** de los campos. Podría haber campos nuevos, o algunos de los campos anteriores podrían ya no estar disponibles. Debido a la **Estructura dinámica**, habilitada, el filtro identifica automáticamente la estructura del archivo recibido. Al mismo tiempo, el filtro lee los nombres de campos y sus valores (pares **name : value**) de los datos. Esto elimina la necesidad de asignación manual de campos a variables.

[Usar filtro de datos](#) la acción no ofrece posibilidades de asignación, debido a que realiza la asignación en forma dinámica. Incluso no tendrá que definir las variables de etiquetas en la configuración de los desencadenadores. La acción asignará valores de campo a las variables de las etiquetas del mismo nombre, sin tener que importar las variables de la etiqueta. No obstante, esta regla se aplica solamente a la acción [Imprimir etiqueta](#). Si quiere usar los valores de campo en otra acción, deberá definir variables en el desencadenador, y mantener, al mismo tiempo, la asignación automática de *variable a campo*.



### NOTA

No se producirá ningún error si el campo disponible en los datos de entrada no tiene una variable de etiqueta correspondiente. ignora en silencio las variables faltantes.



## Configurar área de asignación

El área de asignación se configura usando el mismo procedimiento que la subárea. Para obtener más información, consulte el tema [Definir subáreas](#). El área de asignación puede definirse al nivel de raíz de los datos, donde aparece una sola vez. También se puede configurar dentro de una subárea, de modo que se ejecute para cada bloque de datos en la subárea.

## Configurar campos en el área de asignación

Cuando cree una área de asignación, el filtro define automáticamente los dos marcadores de posición. Estos dos marcadores de posición definen el par `nombre:Valor`.

- **Nombre de variable:** Especifica el campo, cuyo contenido será el nombre de la variable (componente `nombre` en un par). Configure el campo usando el mismo procedimiento que con los campos de documentos. Para obtener más información, consulte el tema [Definir campos](#).
- **Valor de variable:** Especifica el campo cuyo contenido será el valor de la variable (componente `valor` en un par). Configure el campo usando el mismo procedimiento que con los campos de documentos. Para obtener más información, consulte el tema [Definir campos](#).

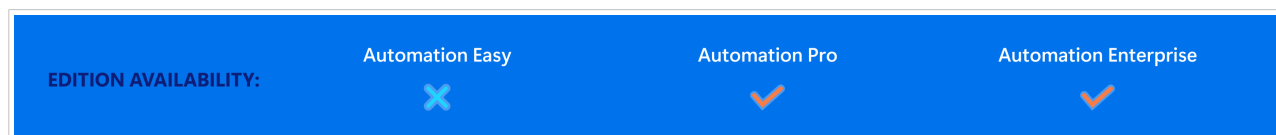
## Ejemplo

El área entre ^XA y ^XZ es el área de asignación. Cada línea del área de asignación proporciona el par **nombre:valor**. **Nombre** El nombre se define como valor entre el sexto carácter de la línea y el carácter de igual. **Valor** El valor se define como valor entre el carácter de igual y el final de la línea con desplazamiento negativo de tres caracteres.

```
^XA
^FD01DonationHR=G095605 3412625^FS
^FD02DonationBC=DG0956053412625^FS
^FD03HospitalNoHR=HN060241^FS
^FD04HospitalNoBC=060241^FS
^FD05Surname=Hawley^FS
^FD07Forename=Annie^FS
^FD09Product=Blood^FS
^FD10PatientBlGp=O Rh +ve^FS
^FD11DoB=27 June 1947^FS
^FD12DateReqd=25 Dec 2012^FS
^XZ
```

## 2.3. Configurar filtro XML

### 2.3.1. Filtro XML



Para saber más sobre los filtros en general, consulte el tema [Comprender los filtros](#).

Utilice este filtro siempre que un desencadenador reciba los datos en código XML. El filtro le permite extraer campos individuales, campos en las subáreas reiterativas, e incluso pares **nombre-valor**. La estructura XML define elemento y subelementos, atributos y sus valores, y valores de texto (valores de elementos).

A pesar de que se puede definir la estructura del archivo XML, NiceLabel recomienda importar la estructura del archivo XML de muestra existente. Haga clic en el botón **Importar estructura de datos** en la banda de opciones. Después de importar la estructura XML, la sección Vista previa de datos muestra el contenido XML. También resalta los elementos y atributos que usted define como campos de salida.

Para ejemplos de datos XML, consulte la sección [Datos XML](#).

## Definir estructuras

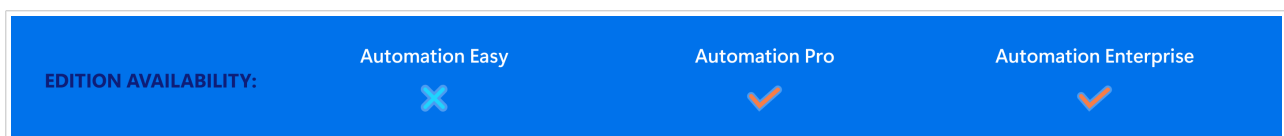
Para usar los elementos XML, debe configurar su uso como:

- **Valor de variable:** Especifique que quiere usar el elemento seleccionado como campo y que asignará su valor a las variables respectivas en la acción [Usar filtro de datos](#). Para obtener más información, consulte el tema [Definir campos XML](#).
  - **Elemento opcional:** Especifique que este elemento no es obligatorio. Esto se corresponde con el atributo del esquema XML (archivo XSD) `minOccurs=0`. La variable asignada a dicho campo tendrá un valor vacío, cuando el elemento no aparezca en XML.
- **Bloque de datos:** Especifica que el elemento seleccionado se produce múltiples veces y proporciona datos para una única etiqueta. El bloque de datos se puede definir como área reiterativa, como área de asignación, o ambas.
  - **Área reiterativa:** Especifique que quiere extraer valores de todos los bloques de datos reiterativos, no solo el primero. Puede definir campos dentro de cada bloque de datos. Debe asignar los campos definidos a las variables respectivas en la acción [Usar filtro de datos](#). Para obtener más información, consulte el tema [Definir elementos reiterativos](#).
  - **Área de asignación:** Especifique que el bloque de datos tiene pares `nombre-valor`. Los nombres de los campos y sus valores se leen simultáneamente. La asignación de variables se realiza automáticamente. Use esta función para ajustar el filtro a los datos de entrada cambiantes, lo que elimina el tiempo de mantenimiento. Para obtener más información, consulte el tema [Definir área de asignación XML](#).

La sección Vista previa de los datos simplifica la configuración. El resultado de una regla de filtro definida se resaltarán en el área de vista previa.

Para cambiar los datos XML en vista previa, haga clic en **Abrir** y busque un nuevo archivo XML de muestra.

## 2.3.2. Definir campos XML



Cuando define campos XML, hace que los valores de los elementos seleccionados estén disponibles automáticamente. La definición de filtros hace que dichos filtros estén disponibles para asignación a variables en acciones. Esto le permite extraer valores de elementos o atributos.

Para definir el valor de elemento, realice lo siguiente:

1. Seleccione el elemento o el atributo en la lista de estructuras.
2. Para **Uso** seleccione **Valor de variable**.

3. El elemento se muestra en la lista de estructuras con letras en negrita, indicando que está en uso.
4. El elemento o nombre del atributo se utilizará como el nombre de campo de salida.
5. En la sección Vista previa de los datos se resaltará el valor del elemento seleccionado.

The screenshot displays the SAP XML Filter configuration window, divided into three main sections:

- Settings:** Contains a tree view of XML elements. The 'Structure' tab is active, showing a hierarchy where 'MATNR' is selected and highlighted in blue. Other elements like 'LBL\_NAME', 'LBL\_PRINTER', and 'LBL\_QUANTITY' are also visible.
- Element properties:** Shows configuration for the selected 'MATNR' element. The 'Usage' is set to 'Variable value', the 'Path' is '/abap/values/NICELABEL\_JOB/IT\_LABEL\_DATA/item', and the 'Name' is 'MATNR'. A 'Show formatting options' button is present.
- Data Preview:** Displays the XML output for the file 'label\_goods\_receipt.xml'. The XML content is shown with line numbers. The 'MATNR' values from the source XML are highlighted in red in the preview, corresponding to the selected element in the settings.

Two callout boxes provide additional context:

- A box pointing to the 'MATNR' element in the structure tree states: "All highlighted XML elements have been defined as 'Variable value'".
- A box pointing to the XML preview states: "Live preview of execution of filter rules. All highlighted data will be used in MATNR field, each 'item' element providing data for a new label."

## Opciones de formato

Esta sección define las funciones de manipulación de cadenas que se aplicarán a las variables o campos seleccionados. Puede seleccionar una o varias funciones. Las funciones se aplican en el mismo orden seleccionado en la interfaz del usuario: de arriba a abajo.

- **Borrar espacios en el comienzo:** borra todos los caracteres de espacio (código 32 ASCII decimal) en el comienzo de la cadena.
- **Borrar espacios en el final:** borra todos los caracteres de espacio (valor 32 ASCII decimal) en el final de una cadena.
- **Borrar el carácter de apertura y cierre:** borra los caracteres de apertura y cierre seleccionados que aparecen por primera vez en la cadena.

### Ejemplo

Si utiliza "{" para el carácter de apertura y "}" para el carácter de cierre, la cadena de entrada `{{selección}}` se convertirá en `{selección}`.

- **Buscar y reemplazar:** ejecuta la función estándar de buscar y reemplazar a partir de los valores provistos para *buscar qué* y *reemplazar con*. También puede usar expresiones comunes.



#### NOTA

Existen varias implementaciones de expresiones comunes en uso. Utiliza la sintaxis de .NET Framework para las expresiones comunes. Para obtener más información, consulte [Artículo de base de conocimiento](#).

- **Reemplazar caracteres que no se pueden imprimir con espacios:** reemplaza todos los caracteres de control en la cadena con espacios (código 32 ASCII decimal). Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Borrar caracteres que no se pueden imprimir:** borra todos los caracteres de control de la cadena. Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Descodificar caracteres especiales:** descodifica los caracteres (o códigos de control) que no están disponibles en el teclado, como Retorno de carro y Avance de línea. Utiliza una notación para codificar dichos caracteres en lenguaje natural, como `<CR>` para Retorno de carro y `<LF>` para Avance de línea. Para obtener más información, consulte el tema [Introducción de caracteres especiales \(Códigos de control\)](#).

Esta opción convierte caracteres especiales de la sintaxis en caracteres binarios reales.

### Ejemplo

Cuando reciba los datos "`<CR><LF>`", los usa como una cadena simple de 8 caracteres. Deberá habilitar esta opción para interpretar y usar los datos recibidos como dos caracteres binarios `CR` (Retorno de carro – ASCII código 13) y `LF` (Avance de línea – ASCII código 10).

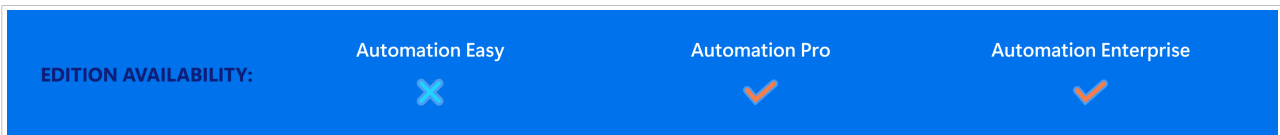
- **Buscar y eliminar todo antes de:** Busca la cadena provista y borra todos los caracteres desde el comienzo de los datos hasta la cadena. La cadena buscada también se puede borrar.
- **Buscar y eliminar todo después de:** Busca la cadena provista y borra todos los caracteres desde la cadena hasta el fin de los datos. La cadena buscada también se puede borrar.
- **Cambiar grafía:** Cambia todos los caracteres de las cadenas a mayúsculas o minúsculas.

### Vista previa de los datos

Esta sección proporciona la vista previa de la definición del campo. Cuando se selecciona el elemento definido, se resalta su posición en los datos en vista previa.

- **Vista previa del nombre del archivo:** especifica el archivo que contiene los datos de muestra que se analizarán a través del filtro. Se copia el archivo en vista previa de la definición del filtro. Si cambia el nombre del archivo en vista previa, se guarda el nuevo nombre.
- **Abrir:** selecciona otro archivo sobre el cual ejecutar las reglas de filtro.
- **Actualizar:** vuelve a ejecutar las reglas de filtro sobre el contenido del nombre del archivo en vista previa. Automation actualiza la sección Vista previa con el resultado.

### 2.3.3. Definir elementos reiterativos en el filtro XML



Si un elemento XML aparece en datos XML múltiples veces, es un elemento reiterativo. Generalmente, el elemento reiterativo tiene datos para una sola etiqueta. Para indicar que quiere utilizar datos de todos los elementos reiterativos, no solo del primero, tiene que definir el elemento como **Bloque de datos** y activar la opción **Elemento reiterativo**. Cuando el filtro contiene la definición de elementos definidos como bloque de datos / elemento reiterativo, la acción [Usar filtro de datos](#) mostrará elementos reiterativos con marcadores de posición anidados. Toda acción anidada debajo de dicho marcador de posición se ejecutará únicamente para los bloques de datos en este nivel.



## Ejemplo

El elemento `<item>` está definido como **Bloque de datos** y **Elemento reiterativo**. Esto indica al filtro que extraiga el elemento `<item>` todas las veces que aparezca, no solo la primera. En este caso, el elemento `<item>` se definirá como el subnivel en la acción **Usar filtro de datos**. Debe anidar las acciones Abrir etiqueta e Imprimir etiqueta debajo del marcador de posición de este subnivel, de modo que esté en bucle todas las veces que aparezca el elemento `<item>`. Como se muestra en el siguiente ejemplo, en tres oportunidades.

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
  <asx:values>
    <NICELABEL_JOB>
      <T IMEST AMP>20130221100527.788134</T IMEST AMP>
      <USER>PGRI</USER>
      <IT _LABEL_DAT A>
        <item>
          <LBL_NAME>goods_r eceipt.nlbl</LBL_NAME>
          <LBL_PRINT ER>Pr oduction01</LBL_PRINT ER>
          <LBL_QUANT IT Y>1</LBL_QUANT IT Y>
          <MAKT X>MASS ONE</MAKT X>
          <MAT NR>28345</MAT NR>
          <MEINS>KG</MEINS>
          <WDAT U>19.01.2012</WDAT U>
          <QUANT IT Y>1</QUANT IT Y>
          <EXIDV>012345678901234560</EXIDV>
        </item>
        <item>
          <LBL_NAME>goods_r eceipt.nlbl</LBL_NAME>
          <LBL_PRINT ER>Pr oduction01</LBL_PRINT ER>
          <LBL_QUANT IT Y>1</LBL_QUANT IT Y>
          <MAKT X>MASS T WO</MAKT X>
          <MAT NR>28346</MAT NR>
          <MEINS>KG</MEINS>
          <WDAT U>11.01.2011</WDAT U>
          <QUANT IT Y>1</QUANT IT Y>
          <EXIDV>012345678901234577</EXIDV>
        </item>
        <item>
          <LBL_NAME>goods_r eceipt.nlbl</LBL_NAME>
          <LBL_PRINT ER>Pr oduction01</LBL_PRINT ER>
          <LBL_QUANT IT Y>1</LBL_QUANT IT Y>
          <MAKT X>MASS T HREE</MAKT X>
          <MAT NR>27844</MAT NR>
          <MEINS>KG</MEINS>
        </item>
      </IT _LABEL_DAT A>
    </NICELABEL_JOB>
  </asx:values>
</asx:abap>
```

```

<WDAT U>07.03.2009</WDAT U>
<QUANT IT Y>1</QUANT IT Y>
<EXIDV>012345678901234584</EXIDV>
</item>

</IT _LABEL_DAT A>
</NICELABEL_JOB>
</asx:values>
</asx:abap>

```



## AVISO

El filtro de datos XML con bloque de datos anidado duplica los valores del área de asignación anterior si una variable está definida en el primer bloque pero no en el segundo. Por ejemplo, se imprimen dos etiquetas. En la primera área de asignación, usted define una variable `<manufacturer_serial>`. En la segunda área de asignación, esta variable no está presente.

```

<?xml ver sion="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" ver sion="1.0">
  <jobs>
    <element>
      <data>
        <item>
          <manufacturer_serial>706583012001</
manufacturer_serial>
          <package_code>AA792CY</package_code>
        </item>
      </data>
      <label_filename>serial_number_label.nlbl</label_filename>
      <print_quantity>1</print_quantity>
      <printer_name>Warehouse_printer_09</printer_name>
    </element>
    <element>
      <data>
        <item>
          <package_code>BE273XS</package_code>
        </item>
      </data>
      <label_filename>serial_number_label.nlbl</label_filename>
      <print_quantity>1</print_quantity>
      <printer_name>Warehouse_09</printer_name>
    </element>
  </jobs>
</asx:abap>

```

El resultado esperado sería la segunda etiqueta sin los datos de `<manufacturer_serial>`. Pero en la impresión real, la segunda etiqueta contendría el valor de `<manufacturer_serial>` de la primera etiqueta, que es 706583012001.

Hay cuatro soluciones posibles:

- Evitar los bloques de datos anidados.
- Definir los valores de las variables como valores vacíos si las variables no se utilizan en los bloques de datos anidados.

```
<item>
  <manufacturer_serial></manufacturer_serial>
  <package_code>AA792CY</package_code>
</item>
```

- Utilizar condiciones en las acciones para establecer variables.
- Utilizar la manipulación de cadenas cuando se detectan las variables faltantes.

### 2.3.4. Definir área de asignación XML

EDITION AVAILABILITY:	Automation Easy ✗	Automation Pro ✓	Automation Enterprise ✓
-----------------------	----------------------	---------------------	----------------------------

El filtro XML identifica automáticamente los campos y sus valores en los datos recibidos. Esto elimina la necesidad de asignación manual *variable-a-campo*.

La funcionalidad **Estructura dinámica** es útil si el desencadenador recibe datos con una estructura cambiante. En dichos casos, la estructura principal de los datos permanece sin cambios (p.ej. los campos están delimitados por coma), o conserva la misma estructura, pero cambia **el orden o el número** de los campos. Podría haber campos nuevos, o algunos de los campos anteriores podrían ya no estar disponibles. Debido a la **Estructura dinámica**, habilitada, el filtro identifica automáticamente la estructura del archivo recibido. Al mismo tiempo, el filtro lee los nombres de campos y sus valores (pares **name:value**) de los datos. Esto elimina la necesidad de asignación manual de campos a variables.

[Usar filtro de datos](#) la acción no ofrece posibilidades de asignación, debido a que realiza la asignación en forma dinámica. Incluso no tendrá que definir las variables de etiquetas en la configuración de los desencadenadores. La acción asignará valores de campo a las variables de las etiquetas del mismo nombre, sin tener que importar las variables de la etiqueta. No obstante, esta regla se aplica solamente a la acción [Imprimir etiqueta](#). Si quiere usar los valores de campo en otra acción, deberá definir variables en el desencadenador, y mantener, al mismo tiempo, la asignación automática de *variable a campo*.



## NOTA

No se producirá ningún error si el campo disponible en los datos de entrada no tiene una variable de etiqueta correspondiente. ignora en silencio las variables faltantes.

The element 'item' is defined as Data block.

The element 'item' is also defined as Assignment area, where the positions of variable names and values are defined.

Live preview of filter rules. Highlighted are values of variables. Names of variables are defined by the XML element names.

## Configurar área de asignación XML

Cuando configura el Bloque de datos como área de asignación, aparecen dos marcadores de posición debajo de la definición de este elemento. Tiene que definir el valor y el nombre del campo, para que el filtro pueda extraer el par **nombre-valor**.

- **Nombre de variable:** Especifica el elemento que tiene el nombre del campo. El nombre se puede definir según el nombre del elemento, el valor del atributo seleccionado o el valor del elemento. Para habilitar la asignación automática, la variable de etiqueta debe tener el mismo nombre.
- **Valor de variable:** Especifica el elemento que tiene el valor del campo. El nombre se puede definir según el nombre del elemento, el valor del atributo seleccionado o el valor del elemento.



## AVISO

El elemento XML que tiene el par **nombre:valor** no puede ser el elemento de raíz, sino que debe tener al menos un elemento de segundo nivel. Por ejemplo, en el ejemplo de XML de abajo, el elemento `<label1>` es de segundo nivel y puede tener el par **nombre:valor**.

## Opciones de formato

Esta sección define las funciones de manipulación de cadenas que se aplicarán a las variables o campos seleccionados. Puede seleccionar una o varias funciones. Las funciones se aplican en el mismo orden seleccionado en la interfaz del usuario: de arriba a abajo.

- **Borrar espacios en el comienzo:** borra todos los caracteres de espacio (código 32 ASCII decimal) en el comienzo de la cadena.
- **Borrar espacios en el final:** borra todos los caracteres de espacio (valor 32 ASCII decimal) en el final de una cadena.
- **Borrar el carácter de apertura y cierre:** borra los caracteres de apertura y cierre seleccionados que aparecen por primera vez en la cadena.

### Ejemplo

Si utiliza "{" para el carácter de apertura y "}" para el carácter de cierre, la cadena de entrada `{{selección}}` se convertirá en `{selección}`.

- **Buscar y reemplazar:** ejecuta la función estándar de buscar y reemplazar a partir de los valores provistos para *buscar qué* y *reemplazar con*. También puede usar expresiones comunes.



#### NOTA

Existen varias implementaciones de expresiones comunes en uso. utiliza la sintaxis de .NET Framework para las expresiones comunes. Para obtener más información, consulte [Artículo de base de conocimiento](#).

- **Reemplazar caracteres que no se pueden imprimir con espacios:** reemplaza todos los caracteres de control en la cadena con espacios (código 32 ASCII decimal). Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Borrar caracteres que no se pueden imprimir:** borra todos los caracteres de control de la cadena. Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Descodificar caracteres especiales:** descodifica los caracteres (o códigos de control) que no están disponibles en el teclado, como Retorno de carro y Avance de línea. utiliza una notación para codificar dichos caracteres en lenguaje natural, como `<CR>` para Retorno de carro y `<LF>` para Avance de línea. Para obtener más información, consulte el tema [Introducción de caracteres especiales \(Códigos de control\)](#).

Esta opción convierte caracteres especiales de la sintaxis en caracteres binarios reales.

### Ejemplo

Cuando reciba los datos "`<CR><LF>`", los usa como una cadena simple de 8 caracteres. Deberá habilitar esta opción para interpretar y usar los datos recibidos como dos caracteres binarios `CR` (Retorno de carro – ASCII código 13) y `LF` (Avance de línea – ASCII código 10).

- **Buscar y eliminar todo antes de:** Busca la cadena provista y borra todos los caracteres desde el comienzo de los datos hasta la cadena. La cadena buscada también se puede borrar.
- **Buscar y eliminar todo después de:** Busca la cadena provista y borra todos los caracteres desde la cadena hasta el fin de los datos. La cadena buscada también se puede borrar.
- **Cambiar grafía:** Cambia todos los caracteres de las cadenas a mayúsculas o minúsculas.

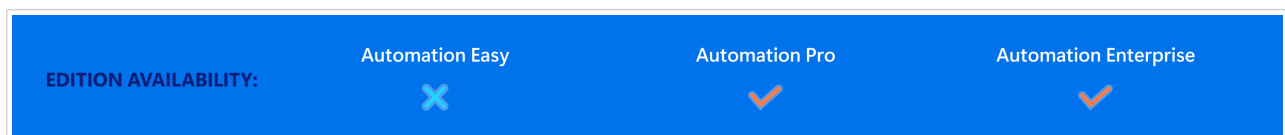
### Ejemplo

El elemento `<label>` está definido como bloque de datos y área de asignación. El **nombre de variable** se define según el valor del nombre del atributo, **el valor de variable** se define según el texto del elemento.

```
<?xml version="1.0" standalone="no"?>
<labels _FORMAT="case.nlbl" _PRINTERNAME="Production01" _QUANTITY="1">
  <label>
    <variable name="CASEID">0000000123</variable>
    <variable name="CARTONTYPE" />
    <variable name="ORDERKEY">0000000534</variable>
    <variable name="BUYERPO" />
    <variable name="ROUTE" > </variable>
    <variable name="CONTAINERDETAILID">0000004212</variable>
    <variable name="SERIALREFERENCE">0</variable>
    <variable name="FILTERVALUE">0</variable>
    <variable name="INDICATORDIGIT">0</variable>
    <variable name="DATE">11/19/2012 10:59:03</variable>
  </label>
</labels>
```

## 2.4. Configurar filtro JSON

### 2.4.1. Filtro JSON



Para saber más sobre los filtros en general, consulte el tema [Comprender los filtros](#).

Utilice el filtro JSON siempre que un desencadenador reciba los datos en código JSON. El filtro JSON le permite usar variables y valores de su archivo JSON. El filtro admite la extracción de datos de matrices JSON.



## NOTA

Automation le permite usar todos los tipos de datos JSON. Lea más sobre los tipos de datos JSON disponibles [aquí](#).

Si bien puede definir la estructura de archivos JSON manualmente, NiceLabel recomienda importar la estructura de los archivos JSON recibidos.

Para importar la estructura de archivos JSON:

1. Vaya a **Filtros de datos** y **Edite** su filtro JSON.
2. Haga clic en **Estructura > Importar estructura de datos**. Navegue hasta su archivo JSON y haga clic **Abrir**.  
Después de importar los archivos JSON, la sección **Vista previa de datos** muestra el contenido JSON. La Vista previa de datos también resalta los elementos que define como campos de salida.

Para ejemplos de datos JSON, consulte la sección [Datos JSON](#).

## Definir estructuras

Para usar los elementos JSON, debe configurar su uso como:

- **Valor de variable:** Especifica que desea utilizar el elemento seleccionado como campo. Al construir la configuración, asigna valores manualmente con las variables respectivas en la acción [Usar filtro de datos](#). Para obtener más información, consulte el tema [Definir campos JSON](#).
  - **Elemento opcional:** Especifique que este elemento no es obligatorio. La variable asignada a dicho campo tendrá un valor vacío cuando el elemento no aparezca en el archivo JSON.
- **Bloque de datos:** Especifica que los subelementos incluidos se producen varias veces y proporcionan datos para sus etiquetas. El bloque de datos se puede definir como área reiterativa, como área de asignación, o ambas. En términos de JSON, el bloque de datos funciona como una matriz.
  - **Área reiterativa:** Especifique que quiere extraer valores de todos los bloques de datos reiterativos, no solo el primero. Puede definir campos dentro de cada bloque de datos. Asigne los campos definidos manualmente a las variables respectivas en la acción [Usar filtro de datos](#). Para obtener más información, consulte el tema [Definir elementos reiterativos en el filtro JSON](#).
  - **Área de asignación:** Crea las variables automáticamente y les asigna valores relevantes. Los nombres de los campos y sus valores se leen simultáneamente. La asignación de variables se realiza automáticamente. Use esta función para ajustar el filtro a los datos de entrada cambiantes, lo que elimina el tiempo de mantenimiento. Para obtener más información, consulte el tema [Definir área de asignación JSON](#).

La sección **Vista previa de los datos** simplifica la configuración. El resultado de una regla de filtro definida se resaltarán en el área de vista previa.

Para cambiar los datos JSON en vista previa, haga clic en **Abrir** y busque un nuevo archivo JSON de muestra.

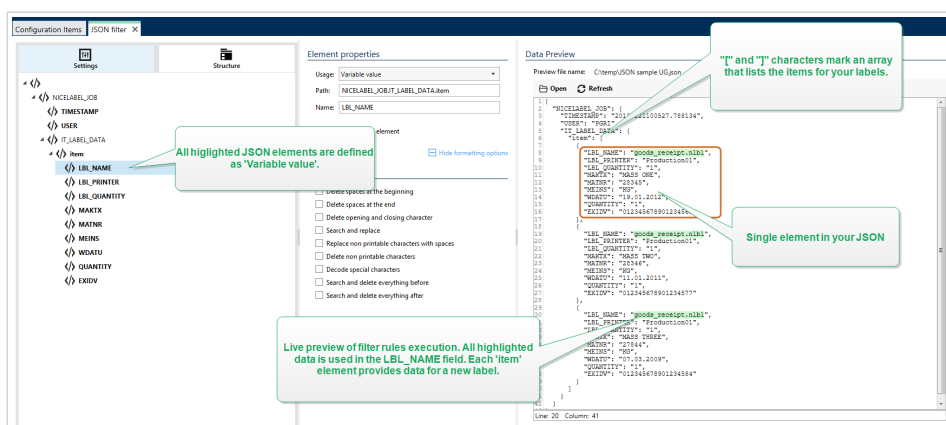
## 2.4.2. Definir campos JSON

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	X	✓	✓

Cuando define campos JSON, hace que los valores de los elementos seleccionados estén disponibles automáticamente. La definición de filtros hace que dichos filtros estén disponibles para asignación a variables en acciones. Esto le permite extraer los valores de los elementos.

Para definir campos JSON:

1. Seleccione el elemento y establezca su **Uso** en **Valor de la variable**.
2. El elemento se muestra en la lista de estructuras con letras en negrita, indicando que está en uso.
3. El elemento se utiliza como el nombre de campo de salida.
4. En la sección Vista previa de los datos se resaltan los valores del elemento seleccionado.





## Opciones de formato

Esta sección define las funciones de manipulación de cadenas que se aplicarán a las variables o campos seleccionados. Puede seleccionar una o varias funciones. Las funciones se aplican en el mismo orden seleccionado en la interfaz del usuario: de arriba a abajo.

- **Borrar espacios en el comienzo:** borra todos los caracteres de espacio (código 32 ASCII decimal) en el comienzo de la cadena.
- **Borrar espacios en el final:** borra todos los caracteres de espacio (valor 32 ASCII decimal) en el final de una cadena.
- **Borrar el carácter de apertura y cierre:** borra los caracteres de apertura y cierre seleccionados que aparecen por primera vez en la cadena.

### Ejemplo

Si utiliza "{" para el carácter de apertura y "}" para el carácter de cierre, la cadena de entrada `{{selección}}` se convertirá en `{selección}`.

- **Buscar y reemplazar:** ejecuta la función estándar de buscar y reemplazar a partir de los valores provistos para *buscar qué* y *reemplazar con*. También puede usar expresiones comunes.



#### NOTA

Existen varias implementaciones de expresiones comunes en uso. utiliza la sintaxis de .NET Framework para las expresiones comunes. Para obtener más información, consulte [Artículo de base de conocimiento](#).

- **Reemplazar caracteres que no se pueden imprimir con espacios:** reemplaza todos los caracteres de control en la cadena con espacios (código 32 ASCII decimal). Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Borrar caracteres que no se pueden imprimir:** borra todos los caracteres de control de la cadena. Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Descodificar caracteres especiales:** descodifica los caracteres (o códigos de control) que no están disponibles en el teclado, como Retorno de carro y Avance de línea. utiliza una notación para codificar dichos caracteres en lenguaje natural, como `<CR>` para Retorno de carro y `<LF>` para Avance de línea. Para obtener más información, consulte el tema [Introducción de caracteres especiales \(Códigos de control\)](#).

Esta opción convierte caracteres especiales de la sintaxis en caracteres binarios reales.

### Ejemplo

Cuando reciba los datos "`<CR><LF>`", los usa como una cadena simple de 8 caracteres. Deberá habilitar esta opción para interpretar y usar los datos recibidos como dos caracteres binarios `CR` (Retorno de carro – ASCII código 13) y `LF` (Avance de línea – ASCII código 10).

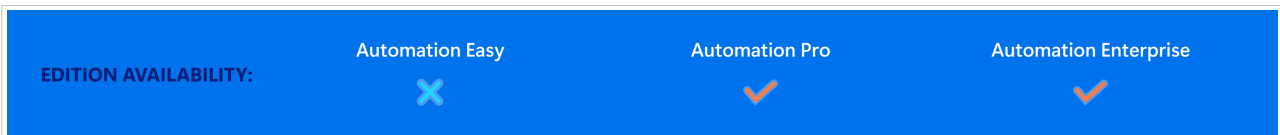
- **Buscar y eliminar todo antes de:** Busca la cadena provista y borra todos los caracteres desde el comienzo de los datos hasta la cadena. La cadena buscada también se puede borrar.
- **Buscar y eliminar todo después de:** Busca la cadena provista y borra todos los caracteres desde la cadena hasta el fin de los datos. La cadena buscada también se puede borrar.
- **Cambiar grafía:** Cambia todos los caracteres de las cadenas a mayúsculas o minúsculas.

### Vista previa de los datos

Esta sección proporciona la vista previa de la definición del campo. Cuando se selecciona el elemento definido, se resalta su posición en los datos en vista previa.

- **Vista previa del nombre del archivo:** especifica el archivo que contiene los datos de muestra que se analizarán a través del filtro. Se copia el archivo en vista previa de la definición del filtro. Si cambia el nombre del archivo en vista previa, se guarda el nuevo nombre.
- **Abrir:** selecciona otro archivo sobre el cual ejecutar las reglas de filtro.
- **Actualizar:** vuelve a ejecutar las reglas de filtro sobre el contenido del nombre del archivo en vista previa. Automation actualiza la sección Vista previa con el resultado.

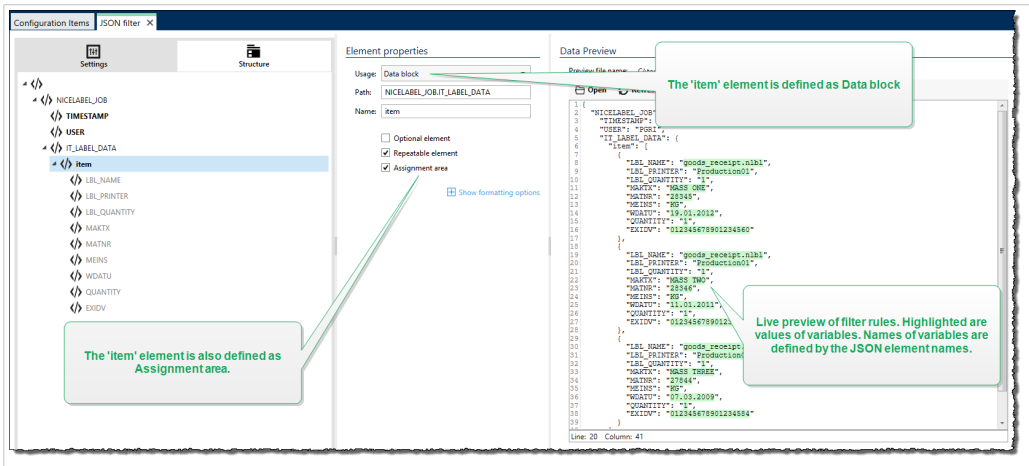
## 2.4.3. Definir elementos reiterativos en el filtro JSON



Si un elemento JSON aparece en datos JSON varias veces, es un elemento reiterativo. Generalmente, el elemento reiterativo tiene datos para una única plantilla de etiquetas. Los elementos reiterativos producen varias etiquetas con datos relevantes.

Para indicar que desea utilizar datos de todos los elementos reiterativos, y no solo del primero:

1. Seleccione el elemento y defínalo como un **Bloque de datos**.
2. Habilite la opción **Elemento reiterativo**.



Cuando el filtro contiene la definición de elementos definidos como bloque de datos / elemento reiterativo, la acción **Usar filtro de datos** mostrará elementos reiterativos con marcadores de posición anidados. Toda acción anidada debajo de dicho marcador de posición se ejecutará únicamente para los bloques de datos en este nivel.

## Ejemplo

El "elemento" se define como **Bloque de datos y Elemento reiterativo**. Esto indica al filtro que extraiga el elemento <item> todas las veces que aparezca en la matriz, no solo la primera. En este caso, el "elemento" se definirá como el subnivel en la acción **Usar filtro de datos**. Debe anidar las acciones Abrir etiqueta e Imprimir etiqueta debajo del marcador de posición de este subnivel, de modo que esté en bucle todas las veces que aparezca el "elemento". Como se muestra en el siguiente ejemplo, en tres oportunidades.

```
{
  "NICELABEL_JOB": {
    "TIMESTAMP": "20130221100527.788134",
    "USER": "PGRI",
    "IT_LABEL_DATA": {
      "item": [
        {
          "LBL_NAME": "goods_receipt.nlbl",
          "LBL_PRINTER": "Production01",
          "LBL_QUANTITY": "1",
          "MAKTX": "MASS ONE",
          "MATNR": "28345",
          "MEINS": "KG",
          "WDATU": "19.01.2012",
          "QUANTITY": "1",
          "EXIDV": "012345678901234560"
        },
        {
          "LBL_NAME": "goods_receipt.nlbl",
          "LBL_PRINTER": "Production01",
          "LBL_QUANTITY": "1",
          "MAKTX": "MASS TWO",
          "MATNR": "28346",
          "MEINS": "KG",
          "WDATU": "11.01.2011",
          "QUANTITY": "1",
          "EXIDV": "012345678901234577"
        },
        {
          "LBL_NAME": "goods_receipt.nlbl",
          "LBL_PRINTER": "Production01",
          "LBL_QUANTITY": "1",
          "MAKTX": "MASS THREE",
          "MATNR": "27844",
          "MEINS": "KG",
          "WDATU": "07.03.2009",
          "QUANTITY": "1",
          "EXIDV": "012345678901234584"
        }
      ]
    }
  }
}
```

```
}  
  }  
    }
```



## AVISO

El filtro de datos JSON con bloque de datos anidado duplica los valores del área de asignación anterior si una variable está definida en el primer bloque pero no en el segundo. Por ejemplo, se imprimen dos etiquetas. En la primera área de asignación, define una variable **"manufacturer\_serial"**. En la segunda área de asignación, esta variable no está presente.

```
{
  "jobs": [
    {
      "data": [
        {
          "manufacturer_serial": "706583012001",
          "package_code": "AA792CY"
        }
      ],
      "label_filename": "serial_number_label.nlbl",
      "print_quantity": 1,
      "printer_name": "Warehouse_printer_09"
    },
    {
      "data": [
        {
          "package_code": "BE273XS"
        }
      ],
      "label_filename": "serial_number_label.nlbl",
      "print_quantity": 1,
      "printer_name": "Warehouse_09"
    }
  ]
}
```

El resultado esperado sería la segunda etiqueta sin los datos de **"manufacturer\_serial"**. Pero en la impresión real, la segunda etiqueta contendría el valor de **"manufacturer\_serial"** de la primera etiqueta, que es 706583012001.

Hay tres soluciones posibles:

- Evitar los bloques de datos anidados.
- Definir los valores de las variables como valores vacíos ("") si las variables no se utilizan en los bloques de datos anidados.

```
"data": [
  {
    "manufacturer_serial": "",
```

```
"package_code" : "AA792CY"
}
```

- Utilizar condiciones en las acciones para establecer variables.
- Utilizar la manipulación de cadenas cuando se detectan las variables faltantes.

## 2.4.4. Definir área de asignación JSON

EDITION AVAILABILITY:	Automation Easy ✗	Automation Pro ✓	Automation Enterprise ✓
-----------------------	----------------------	---------------------	----------------------------

El filtro JSON identifica automáticamente los campos y sus valores en los datos recibidos. Esto elimina la necesidad de asignación manual *variable-a-campo*.

La funcionalidad **Estructura dinámica** es útil si el desencadenador recibe datos con una estructura cambiante. En dichos casos, la estructura principal de los datos permanece sin cambios ( p.ej. los campos están delimitados por coma), o conserva la misma estructura, pero cambia **el orden o el número** de los campos. Podría haber campos nuevos, o algunos de los campos anteriores podrían ya no estar disponibles. Debido a la **Estructura dinámica**, habilitada, el filtro identifica automáticamente la estructura del archivo recibido. Al mismo tiempo, el filtro lee los nombres de campos y sus valores (pares **name:value**) de los datos. Esto elimina la necesidad de asignación manual de campos a variables.

[Usar filtro de datos](#) la acción no ofrece posibilidades de asignación, debido a que realiza la asignación en forma dinámica. Incluso no tendrá que definir las variables de etiquetas en la configuración de los desencadenadores. La acción asignará valores de campo a las variables de las etiquetas del mismo nombre, sin tener que importar las variables de la etiqueta. No obstante, esta regla se aplica solamente a la acción [Imprimir etiqueta](#). Si quiere usar los valores de campo en otra acción, deberá definir variables en el desencadenador, y mantener, al mismo tiempo, la asignación automática de *variable a campo*.



### NOTA

No se producirá ningún error si el campo disponible en los datos de entrada no tiene una variable de etiqueta correspondiente. ignora en silencio las variables faltantes.

The screenshot displays a configuration tool interface. On the left, a tree view shows a hierarchy of configuration items, with 'Item' selected under 'IT\_LABEL\_DATA'. The 'Element properties' panel in the center shows 'Usage' as 'Data block' and 'Name' as 'Item'. It also has checkboxes for 'Optional element', 'Repeatable element', and 'Assignment area', with the last one checked. The 'Data Preview' window on the right shows a JSON array of objects. The first object is highlighted in green, and its 'item' property is also highlighted. A callout box points to the 'item' property, stating: 'The 'Item' element is defined as Data block'. Another callout box points to the highlighted JSON object, stating: 'Live preview of filter rules. Highlighted are values of variables. Names of variables are defined by the JSON element names.' A third callout box points to the 'Assignment area' checkbox, stating: 'The 'Item' element is also defined as Assignment area.'



## NOTA

Debido a que no hay atributos opcionales en JSON, Automation define los **Nombres de las variables** y los **Valores de las variables** automáticamente.



## Opciones de formato

Esta sección define las funciones de manipulación de cadenas que se aplicarán a las variables o campos seleccionados. Puede seleccionar una o varias funciones. Las funciones se aplican en el mismo orden seleccionado en la interfaz del usuario: de arriba a abajo.

- **Borrar espacios en el comienzo:** borra todos los caracteres de espacio (código 32 ASCII decimal) en el comienzo de la cadena.
- **Borrar espacios en el final:** borra todos los caracteres de espacio (valor 32 ASCII decimal) en el final de una cadena.
- **Borrar el carácter de apertura y cierre:** borra los caracteres de apertura y cierre seleccionados que aparecen por primera vez en la cadena.

### Ejemplo

Si utiliza "{" para el carácter de apertura y "}" para el carácter de cierre, la cadena de entrada `{{selección}}` se convertirá en `{selección}`.

- **Buscar y reemplazar:** ejecuta la función estándar de buscar y reemplazar a partir de los valores provistos para *buscar qué* y *reemplazar con*. También puede usar expresiones comunes.



#### NOTA

Existen varias implementaciones de expresiones comunes en uso. utiliza la sintaxis de .NET Framework para las expresiones comunes. Para obtener más información, consulte [Artículo de base de conocimiento](#).

- **Reemplazar caracteres que no se pueden imprimir con espacios:** reemplaza todos los caracteres de control en la cadena con espacios (código 32 ASCII decimal). Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Borrar caracteres que no se pueden imprimir:** borra todos los caracteres de control de la cadena. Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Descodificar caracteres especiales:** descodifica los caracteres (o códigos de control) que no están disponibles en el teclado, como Retorno de carro y Avance de línea. utiliza una notación para codificar dichos caracteres en lenguaje natural, como `<CR>` para Retorno de carro y `<LF>` para Avance de línea. Para obtener más información, consulte el tema [Introducción de caracteres especiales \(Códigos de control\)](#).

Esta opción convierte caracteres especiales de la sintaxis en caracteres binarios reales.

### Ejemplo

Cuando reciba los datos "`<CR><LF>`", los usa como una cadena simple de 8 caracteres. Deberá habilitar esta opción para interpretar y usar los datos recibidos como dos caracteres binarios `CR` (Retorno de carro – ASCII código 13) y `LF` (Avance de línea – ASCII código 10).

- **Buscar y eliminar todo antes de:** Busca la cadena provista y borra todos los caracteres desde el comienzo de los datos hasta la cadena. La cadena buscada también se puede borrar.
- **Buscar y eliminar todo después de:** Busca la cadena provista y borra todos los caracteres desde la cadena hasta el fin de los datos. La cadena buscada también se puede borrar.
- **Cambiar grafía:** Cambia todos los caracteres de las cadenas a mayúsculas o minúsculas.

## Ejemplo

El elemento **"LIST\_ITEM"** está definido como bloque de datos y área de asignación.

```
{
  "DELIVERYNOTE": {
    "LIST_CUSTOMER_INFO": {
      "CUSTOMER_INFO": {
        "CUSTOMER_NAME": "Customer A",
        "CUSTOMER_STREET_ADDRESS": "Test St",
        "CUSTOMER_POST_ADDRESS": "1234, Test City",
        "CUSTOMER_NUMBER": "1234",
        "CURRENCY": "EUR",
        "DELIVERY_METHOD": "Express delivery",
        "EDI_INFORMATION": "EDI",
        "ORDER_TYPE": "CSO",
        "ORDER_NUMBER": "123",
      }
    }
  }
}
```

## 2.5. Configurar nombres de etiqueta e impresora desde datos de entrada

Comúnmente, los filtros se utilizan para extraer valores de los datos recibidos y enviarlos a las variables de las etiquetas para imprimir. En ese caso, el nombre de la etiqueta o el nombre de la impresora tiene una codificación rígida en las acciones. Por ejemplo, la acción [Abrir etiqueta](#) codificará en forma rígida el nombre de la etiqueta, y la acción [Configurar impresora](#) codificará en forma rígida el nombre de la impresora. Sin embargo, los datos de entrada también pueden proporcionar *meta-datos*. Estos son los valores que se usan dentro del procesamiento de NiceLabel Automation, pero no se imprimen en la etiqueta, como el nombre de la etiqueta, nombre de la impresora, cantidad de etiquetas, etc.

Para utilizar los valores de campos de metadatos en el proceso de impresión, realice lo siguiente.

1. **Reconfiguración de filtros:** Debe definir los campos nuevos de los datos de entrada para extraer los campos de metadatos también.

2. **Definición de variables:** Debe definir en forma manual las variables que almacenarán los metadatos, no existen en la etiqueta y no se pueden importar. Utilice nombre intuitivos, tales como `LabelName`, `PrinterName` y `Quantity`. Puede usar el nombre de variable que desee.
3. **Reconfiguración de asignación:** Debe configurar en forma manual la acción [Usar filtro de datos](#) para asignar metacampos a las variables nuevas.
4. **Reconfiguración de acción:** Debe volver a configurar la acción **Abrir etiqueta** para abrir la etiqueta específica según la variable `LabelName`, y la acción Configurar impresora para usar la impresora específica según la variable `PrinterName`.

### Ejemplo

El archivo CSV tiene datos de etiquetas, pero también proporciona *metadatos*, como el nombre de la etiqueta, el nombre de la impresora y la cantidad de etiquetas. El filtro de texto estructurado extraerá todos los campos, enviará los valores relacionados con la etiqueta a las variables de la etiqueta y usará los *metadatos* para configurar las acciones Abrir etiqueta, Configurar impresora e Imprimir etiqueta.

```
label_name;label_count;printer_name;art_code;art_name;ean13;weight  
label1.nlbl;1;CAB A3 203DPI;00265012;SAC.PESTO 250G;383860026501;1,1 kg  
label2.nlbl;1;Zebra R-402;00126502;TAGLIOLINI 250G;383860026002;3,0 kg
```

## 3. Comprender los desencadenadores



### INFORMACIÓN DE NIVEL DE PRODUCTO

Esta funcionalidad no está completamente disponible con cada NiceLabel Automation nivel de producto.

NiceLabel Automation es una aplicación basada en eventos que desencadena la ejecución de acciones al hacer cambios en eventos monitoreados. Puede utilizar cualquiera de los desencadenadores disponibles para monitorear cambios en los eventos, como la colocación de archivos en cierta carpeta, la adquisición de datos en un socket TCP/IP específico, los mensajes HTTP y otros. El propósito principal de un desencadenador es detectar cambios en eventos, recuperar los datos proporcionados por el evento y ejecutar acciones.

La mayoría de los desencadenadores escuchan pasivamente el evento monitoreado a producirse. Existen dos excepciones. El **Desencadenador de base de datos** es un desencadenador activo que verifica periódicamente cambios en la base de datos monitoreada. El **Desencadenador de puerto de serie** puede esperar a una conexión de entrada, o bien sondear activamente en busca de datos en intervalos de tiempo específicos.

#### Procesamiento de desencadenadores

En la mayoría de los casos, el desencadenador recibe los datos que deben imprimirse en etiquetas. Una vez que el desencadenador recibe los datos, las acciones se ejecutan en el orden definido de arriba abajo. Los datos recibidos pueden contener valores para los objetos de las etiquetas. Sin embargo, antes de poder utilizar estos valores, debe extraerlos de los datos recibidos y guardarlos en variables. Los filtros definen las reglas de extracción. Cuando se ejecutan, los filtros guardan los datos extraídos en las variables asignadas. Una vez que haya guardado los datos en las variables, puede ejecutar las acciones que utilizarán las variables, como Imprimir etiqueta.

Una vez que se produce un evento, los datos de entrada proporcionados se guardan en un archivo temporal ubicado en la carpeta %temp% del usuario del servicio. La variable interna **DataFileName** hace referencia a la ubicación del archivo temporal. El archivo se elimina cuando el desencadenador completa su ejecución.

## Propiedades de los desencadenadores

Para configurar un desencadenador, debe definir el modo en que aceptará los datos y las acciones que desea ejecutar. Opcionalmente, puede utilizar también variables. Hay tres secciones en la configuración de un desencadenador.

- **Configuración:** Defina los parámetros principales del desencadenador seleccionado. Seleccione el evento que el desencadenador monitoreará en busca de cambios, o bien el canal de comunicación de entrada. La pestaña Configuración le permite seleccionar el motor de programación de la secuencia y las opciones de seguridad. Las opciones disponibles dependen del tipo de desencadenador. Para obtener más información, consulte el tema [Tipos de desencadenadores](#) más abajo.
- **Variables:** Esta sección define las variables que se necesitan dentro del desencadenador. Por lo general, importará variables desde las plantillas de etiquetas, así que puede asignarlas con los campos extraídos desde los datos de entrada. También puede definir las variables que se utilizarán internamente en varias acciones y que no se enviarán a la etiqueta. Para obtener más información, consulte el tema [Variables](#).
- **Acciones:** Esta sección define las acciones que se ejecutarán cuando el desencadenador detecte algún cambio en el evento monitoreado. Las acciones se ejecutan en orden de arriba abajo. Para obtener más información, consulte el tema [Acciones](#).

## Tipos de desencadenadores

- **Desencadenador de archivo:** Monitorea el cambio en el archivo o conjunto de archivos en la carpeta. El contenido del archivo puede analizarse en filtros y utilizarse en acciones.
- **Desencadenador de puerto de serie:** Monitorea la comunicación de entrada en el puerto de serie RS232. El contenido de la secuencia de entrada puede analizarse en filtros y utilizarse en acciones. Los datos también pueden sondearse desde el dispositivo externo en intervalos de tiempo definidos.
- **Desencadenador de base de datos:** Monitorea los cambios en los registros de las tablas de bases de datos SQL. El contenido del conjunto de datos devueltos puede analizarse y utilizarse en acciones. La base de datos se monitorea en intervalos de tiempo definidos. El desencadenador también puede actualizar la base de datos después de que se ejecutan las acciones utilizando las instrucciones `INSERTAR`, `ACTUALIZAR` e `INSERTAR SQL`.
- **Desencadenador de programador:** Ejecuta su desencadenador en intervalos de tiempo programados.
- **Desencadenador de servidor TCP/IP:** Monitorea la secuencia de datos de entrada sin procesar que llegan al socket definido. El contenido de la secuencia de entrada puede analizarse en filtros y utilizarse en acciones. El desencadenador del servidor TCP/IP puede ser bidireccional y usarse para proporcionar comentarios.
- **Desencadenador de cliente TCP/IP:** Convierte su Automation en un cliente de escucha que se conecta a servidores TCP/IP.
- **Desencadenador de servidor HTTP:** Monitorea la secuencia de datos con formato HTTP de entrada que llegan al socket definido. El contenido de la secuencia de entrada puede analizarse en filtros y utilizarse en acciones. Se puede habilitar la autenticación de usuario. Es bidireccional y suministra comentarios.
- **Desencadenador de servicio Web:** Monitorea la secuencia de datos de entrada que llegan al método de servicio Web definido. El contenido de la secuencia de entrada puede analizarse en filtros y utilizarse en acciones. Es bidireccional y suministra comentarios.
- **Desencadenador en la nube:** Captura datos de Loftware Cloud.

## Tratamiento de errores en desencadenadores

- **Errores de configuración:** El desencadenador es un estado de error si se configura de manera errónea o incompleta. Por ejemplo, si configuró el desencadenador de archivo, pero no especificó el nombre del archivo para verificar en busca de cambios. O bien, si definió la acción para imprimir etiquetas, pero no especificó el nombre de la etiqueta. Puede guardar desencadenadores que contengan errores de configuración, pero no podrá ejecutarlos en Automation Manager hasta que resuelva el problema. El error en el nivel inferior en la configuración se propagará hasta el nivel superior, así que es fácil determinar la ubicación del error.

### Ejemplo

Si tiene una acción en estado de error, todas las acciones de nivel superior indican el estado de error. El icono de error se muestra en la pestaña Acciones y en el nombre del desencadenador.

- **Configuraciones superpuestas:** Si bien es perfectamente aceptable que la configuración incluya desencadenadores que monitorean el mismo evento, como el mismo nombre de archivo o la escucha en el mismo puerto TCP/IP, dichos desencadenadores no pueden ejecutarse al mismo tiempo. Cuando inicie el desencadenador en Automation Manager, solo se iniciará si no hay otro desencadenador de esa o de otra configuración monitoreando el mismo evento.

## Comentarios sobre el estado del trabajo de impresión

Consulte la sección [Comentarios sobre el estado del trabajo de impresión](#).

## 3.1. Desencadenador de archivo

Para saber más sobre los desencadenadores en general, consulte el tema [Comprender los desencadenadores](#).

El evento desencadenador de archivo se produce si:

- cambia el archivo monitoreado
- cambia un conjunto de archivos en la carpeta monitoreada
- aparece un nuevo archivo en la carpeta monitoreada

En función de la configuración del desencadenador, el sistema Windows alerta al desencadenador sobre los archivos que se cambiaron, o el desencadenador mismo mantiene una lista de la marca de tiempo de la última escritura de los archivos y se dispara cuando el archivo tiene una marca de tiempo más reciente.



## NOTA

Uso típico: el sistema de negocio ejecuta una transacción, que en efecto genera el archivo desencadenador en la carpeta compartida. El contenido de los datos puede estructurarse en CSV, XML y otros formatos, o bien en un formato heredado. De cualquier modo, NiceLabel Automation leerá los datos, analizará los valores utilizando los filtros y los imprimirá en etiquetas. Para obtener más información sobre el modo de analizar y extraer datos, consulte la sección [Comprender los filtros](#).

## General

Esta sección le permite configurar los parámetros más importantes del desencadenador de archivo.

- **Nombre:** Especifica el nombre único del desencadenador. El nombre lo ayuda a distinguir los diferentes desencadenadores cuando los configura en Automation Builder y luego cuando los ejecuta en Automation Manager.
- **Descripción:** Le permite describir el rol de este desencadenador. Ingrese una breve explicación acerca de lo que hace el desencadenador para ayudar a los usuarios.
- **Detectar el archivo especificado:** Especifica la ruta y el nombre del archivo, el cual está monitoreando en busca de cambios.
- **Detectar conjunto de archivos en una carpeta específica:** Especifica la ruta a la carpeta, la cual está monitoreando en busca de cambios de archivos, y los nombres de archivos. Puede utilizar los comodines estándares de Windows "\*" y "?". Algunos tipos de archivo se predefinen en el cuadro desplegable; usted puede ingresar sus propios tipos.



## NOTA

Al monitorear la carpeta de red, asegúrese de utilizar la notación UNC de `\servidor\compartir\archivo`. Para obtener más información, consulte el tema [Acceso a recursos de red compartidos](#).

- **Detectar cambios automáticamente:** NiceLabel Automation responde a los cambios en los archivos tan pronto como se crea o modifica un archivo. En este caso, el sistema operativo Windows informa al Servicio NiceLabel Automation sobre el cambio. Puede utilizarlo cuando la carpeta monitoreada se encuentra en el disco local y también en algunos entornos de red.
- **Verificar cambios en la carpeta en intervalos:** NiceLabel Automation escanea la carpeta en busca de cambios en el archivo en los intervalos de tiempo definidos. En este caso, NiceLabel Automation monitorea la carpeta en busca de cambios de archivos. Este método de sondeo suele ser más lento que la detección automática. Utilícelo como reserva, cuando la detección automática no puede utilizarse en su entorno.

## Ejecución

Las opciones de la sección **Acceso a archivo** especifican el modo en que la aplicación accederá al archivo desencadenador.



- **Abrir archivo exclusivamente:** Especifica que se abra el archivo desencadenador en modo exclusivo. Ninguna otra aplicación puede acceder al archivo al mismo tiempo. Esta es la opción de selección predeterminada.
- **Abrir archivo con permisos de solo lectura:** Abre el archivo desencadenador en modo de solo lectura.
- **Abrir archivo con permisos de lectura y escritura:** Abre el archivo desencadenador en modo de lectura y escritura.
- **Período de reintento de abrir archivo:** Especifica el período de tiempo después del cual NiceLabel Automation intenta abrir el archivo desencadenador. Si continúa siendo imposible acceder al archivo después de este período, NiceLabel Automation arrojará un error.

Las opciones en la sección **Opciones de monitoreo** especifican las posibilidades de detección de archivos.

- **Comprobar tamaño de archivo:** Habilita la detección de cambios no sólo en la marca de tiempo del archivo, sino también en la longitud del archivo. Los cambios en la marca de tiempo del archivo posiblemente no se detecten en algunos casos. Por lo tanto, Automation también comprueba cambios en el tamaño del archivo y desencadena las acciones.
- **Ignorar archivos desencadenadores vacíos:** Si el archivo desencadenador no tiene contenido, se ignora. Las acciones no se ejecutan.
- **Eliminar archivo desencadenador:** Una vez que el cambio en el archivo desencadenador ha sido detectado, y se dispara el desencadenador, Automation elimina el archivo. Al habilitar esta opción, la carpeta se mantendrá limpia de archivos ya procesados.



## NOTA

NiceLabel Automation siempre crea una copia de seguridad de los datos recibidos por el desencadenador. Los contenidos del archivo desencadenador se guardan usando un nombre del archivo único. Esto es importante cuando se necesita el contenido del archivo desencadenador en alguna de las acciones, como **Ejecutar archivo de comando**. La variable *DataFileName* hace referencia a la ubicación de la copia de seguridad de los datos desencadenadores.

- **Vaciar contenidos de archivo:** Una vez que se ejecutan las acciones, el archivo desencadenador se vacía. Esto es útil cuando las aplicaciones de terceros anexan datos al archivo desencadenador. Usted desea conservar el archivo para que pueda anexarse, pero no desea imprimir datos antiguos.
- **Rastrear cambios con el desencadenador inactivo:** Dispara el desencadenador cuando los archivos cambian con el desencadenador inactivo. Cuando su NiceLabel Automation no se implementa en el entorno de alta disponibilidad con servidores de respaldo, los archivos desencadenadores de entrada pueden perderse si el servidor está apagado. Cuando el NiceLabel Automation está nuevamente encendido, los archivos desencadenadores existentes pueden procesarse.
- **Cantidad de ejecuciones de acciones simultáneas:** Especifique la cantidad de ejecuciones de acciones simultáneas. No es necesario esperar a que termine la ejecución para que comience la siguiente. Su orden de procesamiento de acción se mantiene igual mientras, simultáneamente, la misma acción de otro subproceso se comienza a ejecutar.

La cantidad máxima de ejecuciones de acciones simultáneas depende del rendimiento de su hardware. Obtenga más información sobre [Sección 7.1, "Procesamiento paralelo"](#).

## Otros

Las opciones en la sección **Comentarios del motor de impresión** especifican los parámetros de comunicación que le permiten recibir comentarios del motor de impresión.

<b>EDITION AVAILABILITY:</b>	Automation Easy	Automation Pro	Automation Enterprise
	✗	✓	✓

- **Impresión supervisada:** Activa el modo de impresión sincronizada. Utilícela siempre que quiera enviar el estado del trabajo de impresión a una aplicación de terceros. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

Las opciones en la sección **Procesamiento de datos** especifican si quiere recortar los datos para que entren en la variable o ignorar las variables de las etiquetas faltantes. En forma predeterminada, notificará los errores e interrumpirá el proceso de impresión si intenta guardar valores muy largos en la variable de la etiqueta o intenta configurar valores para variables de etiquetas no existentes.

- **Ignorar exceso de contenido de variables:** trunca los valores de datos que exceden la longitud de la variable tal lo definido en el diseñador de etiquetas para que se ajusten. Esta opción está en efecto cuando se configuran valores de variables en los filtros, desde los archivos de comandos y cuando se configuran valores de variables de desencadenadores para las variables de etiquetas del mismo nombre.

## Ejemplo

La variable de la etiqueta acepta 5 caracteres como máximo. Con esta opción habilitada, cualquier valor de más de cinco caracteres se trunca a los primeros cinco caracteres. Si el valor es 1234567 ignora los dígitos 6 y 7.

- **Ignorar variables de etiquetas faltantes:** Al imprimir con [archivos de comandos](#) (como un archivo JOB), el proceso de impresión ignora todas las variables que están:
  - especificadas en el archivo de comandos (que usan el comando [SET](#))
  - no definidas en la etiqueta

Lo mismo sucede si define el área de asignación en un filtro para que extraiga todos los pares nombre-valor, pero su etiqueta contiene menos variables.

Al configurar valores de variables de etiquetas no existentes, informo un error. Si esta opción está habilitada la impresión continúa.

Las opciones en la sección **Secuencia de comandos** especifican las posibilidades de las secuencias de comandos.

- **Idioma de secuencias:** Selecciona el idioma de secuencias para el desencadenador. Todas las acciones **Ejecutar secuencia de comandos** que utiliza dentro de un solo desencadenador usan el idioma de secuencia de comandos. seleccionado.

Las opciones en la sección **Guardar datos recibidos** especifican los comandos disponibles para los datos recibidos por el desencadenador.

- **Guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción y guarde los datos recibidos por el desencadenador. La opción **Variable** habilita un nombre de archivo variable. Seleccione una variable que contenga la ruta y el nombre de archivo.



#### NOTA

NiceLabel Automation guarda los datos recibidos en un archivo temporal. Este archivo temporal se elimina inmediatamente después de que se completa la ejecución del desencadenador. La variable interna `DataFileName` hace referencia a ese nombre de archivo. Para obtener más información, consulte el tema [Variables internas](#).

- **Al guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción para guardar los datos en el desencadenador sólo si se produce un error durante la ejecución de la acción. Puede activar esta opción para mantener los datos que causaron el problema listos para resolver el problema más adelante.



#### NOTA

Asegúrese de habilitar el soporte para impresión supervisada. De lo contrario, no puede detectar errores durante la ejecución. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

## Seguridad

- **Bloquear y cifrar desencadenador:** Habilita la protección del desencadenador. Si lo habilita, el desencadenador queda bloqueado y usted ya no puede editarlo. Esto cifra las acciones. Únicamente el usuario que tenga una contraseña puede desbloquear el desencadenador y modificarlo.

## 3.2. Desencadenador de puerto de serie

Para saber más sobre los desencadenadores en general, consulte el tema [Comprender los desencadenadores](#).

El evento de desencadenador de puerto de serie se produce cuando se reciben datos en el puerto de serie RS232 monitoreado.

Uso típico: **(1) Reemplazo de impresoras.** Está retirando la impresora de etiquetas conectada al puerto de serie existente. En su lugar, NiceLabel Automation aceptará los datos, extraerá los valores para los objetos de etiqueta desde la secuencia de impresión recibida y creará un trabajo de impresión para el nuevo

modelo de impresora. **(2) Balanzas.** La balanza proporciona los datos sobre el objeto pesado. NiceLabel Automation extrae los datos requeridos de la secuencia de datos recibidos e imprime una etiqueta. Para obtener más información sobre el modo de analizar y extraer datos, consulte la sección [Comprender los filtros](#).

## General

Esta sección le permite configurar los parámetros más importantes del desencadenador de archivo.

- **Nombre:** Especifica el nombre único del desencadenador. El nombre lo ayuda a distinguir los diferentes desencadenadores cuando los configura en Automation Builder y luego cuando los ejecuta en Automation Manager.
- **Descripción:** Le permite describir el rol de este desencadenador. Ingrese una breve explicación acerca de lo que hace el desencadenador para ayudar a los usuarios.
- **Puerto:** Especifica el número del puerto serial (COM) en el que se reciben los datos entrantes. Utilice un puerto que no esté en uso por ninguna otra aplicación o dispositivo, como un controlador de impresora. Si el puerto seleccionado está en uso, no podrá iniciar el desencadenador en Automation Manager.

Las opciones de la sección **Configuración de puerto** especifican los parámetros de comunicación que deben coincidir con los parámetros en el dispositivo del puerto de serie.




- **Desactivar inicialización de puerto:** Especifica que la inicialización del puerto no se ejecutará cuando se inicie el desencadenador en Automation Manager. En ocasiones, esta opción se requiere para puertos COM virtuales.

## Ejecución

- **Usar datos de inicialización:** Especifica que usted desea enviar la cadena de inicialización al dispositivo de serie cada vez que se inicia el desencadenador. Algunos dispositivos de serie deben activarse o colocarse en modo de espera antes de que puedan proporcionar datos. Para obtener más información sobre la cadena de inicialización y si esta es necesaria para usted, consulte la guía del usuario del dispositivo. Puede incluir caracteres binarios. Para obtener más información, consulte el tema [Introducir caracteres especiales](#).
- **Usar sondeo de datos:** especifica que el desencadenador solicitará activamente datos al dispositivo. Dentro de los intervalos de tiempo especificados, el desencadenador enviará los comandos proporcionados en el campo **Contenido**. Este campo puede incluir caracteres binarios. Para obtener más información, consulte el tema [Introducir caracteres especiales](#).

## Otros

Las opciones en la sección **Comentarios del motor de impresión** especifican los parámetros de comunicación que le permiten recibir comentarios del motor de impresión.

<b>EDITION AVAILABILITY:</b>	Automation Easy	Automation Pro	Automation Enterprise
			

- **Impresión supervisada:** Activa el modo de impresión sincronizada. Utilícela siempre que quiera enviar el estado del trabajo de impresión a una aplicación de terceros. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

Las opciones en la sección **Procesamiento de datos** especifican si quiere recortar los datos para que entren en la variable o ignorar las variables de las etiquetas faltantes. En forma predeterminada, notificará los errores e interrumpirá el proceso de impresión si intenta guardar valores muy largos en la variable de la etiqueta o intenta configurar valores para variables de etiquetas no existentes.

- **Ignorar exceso de contenido de variables:** trunca los valores de datos que exceden la longitud de la variable tal lo definido en el diseñador de etiquetas para que se ajusten. Esta opción está en efecto cuando se configuran valores de variables en los filtros, desde los archivos de comandos y cuando se configuran valores de variables de desencadenadores para las variables de etiquetas del mismo nombre.

### Ejemplo

La variable de la etiqueta acepta 5 caracteres como máximo. Con esta opción habilitada, cualquier valor de más de cinco caracteres se trunca a los primeros cinco caracteres. Si el valor es 1234567 ignora los dígitos 6 y 7.

- **Ignorar variables de etiquetas faltantes:** Al imprimir con [archivos de comandos](#) (como un archivo JOB), el proceso de impresión ignora todas las variables que están:
  - especificadas en el archivo de comandos (que usan el comando [SET](#))
  - no definidas en la etiqueta

Lo mismo sucede si define el área de asignación en un filtro para que extraiga todos los pares nombre-valor, pero su etiqueta contiene menos variables.

Al configurar valores de variables de etiquetas no existentes, informo un error. Si esta opción está habilitada la impresión continúa.

Las opciones en la sección **Secuencia de comandos** especifican las posibilidades de las secuencias de comandos.

- **Idioma de secuencias:** Selecciona el idioma de secuencias para el desencadenador. Todas las acciones **Ejecutar secuencia de comandos** que utiliza dentro de un solo desencadenador usan el idioma de secuencia de comandos. seleccionado.

Las opciones en la sección **Guardar datos recibidos** especifican los comandos disponibles para los datos recibidos por el desencadenador.

- **Guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción y guarde los datos recibidos por el desencadenador. La opción **Variable** habilita un nombre de archivo variable. Seleccione una variable que contenga la ruta y el nombre de archivo.
- **Al guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción para guardar los datos en el desencadenador sólo si se produce un error durante la ejecución de la acción.

Puede activar esta opción para mantener los datos que causaron el problema listos para resolver el problema más adelante.



#### NOTA

Asegúrese de habilitar el soporte para impresión supervisada. De lo contrario, no puede detectar errores durante la ejecución. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).



#### NOTA

NiceLabel Automation guarda los datos recibidos en un archivo temporal. Este archivo temporal se elimina inmediatamente después de que se completa la ejecución del desencadenador. La variable interna `DataFileName` hace referencia a ese nombre de archivo. Para obtener más información, consulte el tema [Variables internas](#).

### Seguridad

- **Bloquear y cifrar desencadenador:** Habilita la protección del desencadenador. Si lo habilita, el desencadenador queda bloqueado y usted ya no puede editarlo. Esto cifra las acciones. Únicamente el usuario que tenga una contraseña puede desbloquear el desencadenador y modificarlo.

## 3.3. Desencadenador de base de datos

Para saber más sobre los desencadenadores en general, consulte el tema [Comprender los desencadenadores](#).

El evento de desencadenador de base de datos se produce cuando se detecta un cambio en la tabla de la base de datos monitoreada. Puede haber registros nuevos, o registros existentes que se han actualizado. El desencadenador de base de datos no espera a que se produzca ningún cambio de evento, como la entrega de datos. Por el contrario, extrae datos de la base de datos en intervalos de tiempo definidos.

Uso típico: El sistema de negocio existente ejecuta una transacción, que en efecto actualiza los datos en alguna tabla de base de datos. NiceLabel Automation detectará los registros actualizados y nuevos, e imprimirá su contenido en las etiquetas.

### General

Esta sección le permite configurar los parámetros más importantes del desencadenador de archivo.

- **Nombre:** Especifica el nombre único del desencadenador. El nombre lo ayuda a distinguir los diferentes desencadenadores cuando los configura en Automation Builder y luego cuando los ejecuta en Automation Manager.
- **Descripción:** Le permite describir el rol de este desencadenador. Ingrese una breve explicación acerca de lo que hace el desencadenador para ayudar a los usuarios.

- **Conexión a la base de datos:** Especifica la cadena de conexión con la base de datos. Haga clic en **Definir** para abrir el cuadro de diálogo de la base de datos. Úselo para configurar la conexión de la base de datos, incluido el tipo de bases de datos, el nombre de tabla y las credenciales de usuario. Debe conectarse a la base de datos que permita el acceso con comandos SQL. Por ello, no puede utilizar el desencadenador de base de datos para detectar automáticamente cambios de datos en archivos de texto CSV y en hojas de cálculo de Microsoft Excel.



#### NOTA

Los detalles de la configuración dependen del tipo de base de datos seleccionada. Las opciones del cuadro de diálogo dependen del controlador de la base de datos que utilice. Para ver los detalles de configuración, consulte la guía del usuario del controlador de la base de datos. Para obtener más información sobre conectividad de base de datos, consulte la sección [Acceso a bases de datos](#).

- **Comprobar base de datos en los intervalos de tiempo:** Especifica el intervalo de tiempo en el cual se sondea la base de datos para detectar cambios en los registros.
- **Opciones de detección y avanzadas:** Estas opciones le permiten ajustar el mecanismo de detección de registros. Cuando se adquieren registros de la base de datos, la pestaña Acción muestra automáticamente el objeto Para cada registro, donde puede asignar campos de la tabla a variables de etiqueta.

#### Obtener registros basados en el valor único del campo incremental

Con esta opción habilitada, el desencadenador monitorea los campos numéricos autoincrementales especificados en la tabla. NiceLabel Automation recordará el valor del campo para el último registro procesado. En el siguiente intervalo de sondeo, solo se adquirirán los registros con valores mayores que el valor recordado.

Para configurar esta opción, debe seleccionar el nombre de la tabla donde residen los registros (**nombre de tabla**), el campo con incremento automático (**campo clave**) y el valor de inicio para el campo (**valor predeterminado del campo clave**). Internamente, la variable **KeyField** se utiliza para hacer referencia al último valor recordado del campo clave.



#### NOTA

El último valor del campo clave se recuerda internamente, pero no se actualiza en la configuración, de modo que el **valor predeterminado del campo clave** no cambia en este cuadro de diálogo. Puede volver a cargar de manera segura la configuración o detener/iniciar el desencadenador en Automation Manager y, aun así, mantener el último valor recordado. Sin embargo, si elimina la configuración desde Automation Manager y la vuelve a agregar, el valor del último campo clave recordado volverá a ser el que definió como **valor predeterminado del campo clave**.

### Obtener registros y borrarlos

Con esta opción seleccionada, todos los registros se adquieren de la tabla y se eliminan. Para configurar esta opción, debe seleccionar el nombre de la tabla en la que residen los registros (**nombre de tabla**) y especificar la clave principal en la tabla (**campos clave**). Si bien Automation permite tener una tabla sin una clave principal, es muy recomendable definir una. Si la clave principal existe, los registros se eliminarán de a uno cuando cada uno se procese en las acciones.



#### AVISO

Si la clave principal no existe, todos los registros obtenidos en el desencadenador actual se eliminarán a la vez. Eso no es problema si no se produce ningún error durante el procesamiento de registros. Sin embargo, si se produce un error de procesamiento con algún registro, Automation deja de procesar otros registros. Como todos los registros capturados en ese intervalo de sondeo ya se han eliminado sin procesarse, puede perder datos. Por lo tanto, es recomendable tener una clave principal en una tabla.

### Ejemplos de código SQL



#### NOTA

Estas instrucciones SQL son de solo lectura y se presentan únicamente a modo de referencia. Para proporcionar instrucciones SQL personalizadas, seleccione el método de detección **Obtener y administrar registros con SQL personalizadas**.

#### Tabla de ejemplo:

ID	ProductID	CódigoEAN	DescDeProducto	YaImpreso
1	CAS0006	8021228110014	CASONCELLI ALLA CARNE 250G	Y
2	PAS501	8021228310001	BIGOLI 250G	
3	PAS502GI	8021228310018	TAGLIATELLE 250G	

#### Ejemplo de Actualización de instrucción SQL cuando la tabla contiene el índice principal:

```
DELETE FROM [Table]
WHERE [ID] = :ID
```

**ID** en la tabla se define como el índice principal. La construcción **:ID** en la cláusula WHERE contiene el valor de campo ID en cada iteración. Para el primer registro, el valor de **ID** es 1, para el segundo registro es 2, etc. Al colocar dos puntos delante del nombre de campo en la instrucción SQL, se especifica el uso de la variable.

#### Ejemplo de Actualización de instrucción SQL cuando la tabla no tiene el índice principal definido:

```
DELETE FROM [Table]
```



Cuando no haya un índice principal definido en la tabla, todos los registros se eliminan de la tabla cuando el primer registro se haya procesado.

### Obtener registros y actualizarlos

Con esta opción seleccionada, todos los registros se adquieren de la tabla y se actualizan. Puede escribir un valor personalizado en el campo en la tabla para indicar "estos registros ya están impresos". Para configurar esta opción, debe seleccionar el nombre de la tabla en la que residen los registros (**nombre de tabla**), seleccionar el campo que desea actualizar (**actualizar campo**) e ingresar el valor que se almacenará en el campo (**actualizar valor**). Internamente, la variable `UpdateValue` se utiliza en la instrucción SQL para hacer referencia al valor actual del campo (**actualizar valor**).

Si bien Automation permite tener una tabla sin una clave principal, es muy recomendable definir una. Si la clave principal existe, los registros se actualizarán de a uno cuando cada uno se procese en las acciones.



#### AVISO

Si la clave principal no existe, todos los registros obtenidos en el desencadenador actual se actualizarán a la vez. Eso está bien si no hay ningún error de procesamiento de los registros. Pero si hay algún error al procesar un registro, la Automatización detendrá el procesamiento de los demás registros. Como todos los registros capturados en ese intervalo de sondeo ya se han actualizado sin procesarse en acciones, puede perder datos. Por lo tanto, es una buena idea tener una clave principal en una tabla.

### Ejemplos de código SQL



#### NOTA

Estas instrucciones SQL son de solo lectura y se presentan únicamente a modo de referencia. Para proporcionar instrucciones SQL personalizadas, seleccione el método de detección **Obtener y administrar registros con SQL personalizadas**.

#### Tabla de ejemplo:

ID	ProductID	CódigoEAN	DescDeProducto	YaImpreso
1	CAS0006	8021228110014	CASONCELLI ALLA CARNE 250G	Y
2	PAS501	8021228310001	BIGOLI 250G	
3	PAS502GI	8021228310018	TAGLIATELLE 250G	

#### Ejemplo de Actualización de instrucción SQL cuando la tabla contiene el índice principal:

```
UPDATE [Table]
SET [AlreadyPrinted] = :UpdateValue
WHERE [ID] = :ID
```

`ID` en la tabla se define como el índice principal. La construcción `:ID` en la cláusula `WHERE` contiene el valor de campo `ID` en cada iteración. Para el primer registro, el valor de `ID` es 1, para el segundo

registro es 2, etc. Al agregar dos puntos delante del nombre de campo en la instrucción SQL, se especifica el uso de la variable. El campo `UpdateValue` se define en la configuración del desencadenador en el campo de edición **Actualizar valor**.

**Ejemplo de Actualización de instrucción SQL cuando la tabla no tiene el índice principal definido:**

```
UPDATE [Table]
SET [AlreadyPrinted] = :UpdateValue
```

Cuando no haya un índice principal definido en la tabla, todos los registros de la tabla se actualizarán cuando el primer registro se haya procesado.

## Obtener y administrar registros con SQL personalizado

En este caso, la creación de instrucciones SQL para la extracción de registros y actualización de campos depende completamente de usted. Para configurar esta opción, debe proporcionar una instrucción SQL personalizada para adquirir registros (**buscar instrucción SQL**) y para actualizar los registros después del procesamiento (**actualizar instrucción SQL**). Haga clic en el botón **Probar** para hacer una ejecución de prueba de sus instrucciones SQL y ver el resultado en pantalla.

Puede utilizar valores de campos de tabla o valores de variables de desencadenador como parámetros en la cláusula WHERE de la instrucción SQL. Debe colocar dos puntos (:) delante del nombre de campo o variable. Esto indica a NiceLabel Automation que utilice el valor actual de ese campo o variable.

## Ejemplos de código SQL

### Tabla de ejemplo:

ID	ProductID	CódigoEAN	DescDeProducto	YaImpreso
1	CAS0006	8021228110014	CASONCELLI ALLA CARNE 250G	Y
2	PAS501	8021228310001	BIGOLI 250G	
3	PAS502GI	8021228310018	TAGLIATELLE 250G	

### Ejemplo de Buscar instrucción SQL:

Para obtener los registros que aún no han sido impresos, haga lo siguiente. El campo **AlreadyPrinted** no debe contener el valor **Y**, y tener un valor en blanco o NULO.

```
SELECT * FROM Table
WHERE AlreadyPrinted <> 'Y' or AlreadyPrinted is NULL
```

Desde la tabla de muestra a continuación, se extraerán dos registros con valores de ID 2 y 3. El primer registro ya se imprimió y se ignorará.

### Ejemplo de instrucción Actualizar SQL:

Para marcar los registros ya impresos con el valor **Y** en el campo **AlreadyPrinted** haga lo siguiente:

```
UPDATE [Table]
SET [AlreadyPrinted] = 'Y'
WHERE [ID] = :ID
```

Coloque dos puntos (:) delante del nombre de variable en la instrucción SQL para identificarla como variable. Puede utilizar cualquier campo de la tabla para los parámetros en la cláusula WHERE. En el ejemplo, estamos actualizando el campo **AlreadyPrinted** únicamente para el registro que se está procesando actualmente (el valor de campo **ID** debe ser el mismo que el valor del registro actual). Al igual que haría referencia a otros campos en el registro como **:ProductID** o **:CodeEAN**, o incluso a las variables definidas dentro de este desencadenador de base de datos.

Para eliminar el registro actual de la tabla, haga lo siguiente:

```
DELETE FROM [Table]
WHERE [ID] = :ID
```

**Mostrar instrucción SQL:** Expanda esta sección para ver la instrucción SQL generada y escribir su propia instrucción, si ha seleccionado la opción **Obtener y administrar registros con SQL personalizadas**.

### Vista previa de ejecución de SQL

Para probar la ejecución de las sentencias SQL y ver qué efecto tendrían, haga clic en el botón Probar en la barra de herramientas del área de edición SQL. Se abrirá la sección Vista previa de los datos en el panel de la derecha. Haga clic en el botón **Ejecutar** para iniciar el código SQL. Cuando utiliza valores de campo de tabla en la instrucción SQL (con dos puntos [:] delante del nombre de campo), debe proporcionar los valores de prueba para ellos.



#### NOTA

Si tiene abierta la sección Vista previa de los datos y recientemente ha agregado algunas variables a la secuencia, haga clic en el botón **Probar** dos veces. Así se cierra y se abre la sección Vista previa de los datos y se actualiza la lista de variables en la vista previa.

- **Simular ejecución:** Especifica que se ignoren todos los cambios realizados a la base de datos. La transacción de la base de datos se revierte para que no se escriban datos en ella.

### Ejecución

Las opciones en Ejecución especifican cuándo se realizará la actualización de la base de datos. El tipo de actualización depende de las Opciones de detección para el desencadenador.

- **Antes de procesar acciones:** Especifica que los registros se actualizarán antes de que se inicie la ejecución de las acciones definidas para ese desencadenador.
- **Después de procesar acciones:** Especifica que los registros se actualizarán después de que se inicie la ejecución de las acciones definidas para ese desencadenador. Por lo general, conviene actualizar los registros después de que se hayan procesado correctamente.



#### NOTA

De ser necesario, puede también actualizar los registros mientras aún se están ejecutando las acciones. Para obtener más información, consulte el tema [Ejecutar instrucción SQL](#).

### Otros

Las opciones en la sección **Comentarios del motor de impresión** especifican los parámetros de comunicación que le permiten recibir comentarios del motor de impresión.

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✗	✓	✓

- **Impresión supervisada:** Activa el modo de impresión sincronizada. Utilícela siempre que quiera enviar el estado del trabajo de impresión a una aplicación de terceros. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

Las opciones en la sección **Procesamiento de datos** especifican si quiere recortar los datos para que entren en la variable o ignorar las variables de las etiquetas faltantes. En forma predeterminada, notificará los errores e interrumpirá el proceso de impresión si intenta guardar valores muy largos en la variable de la etiqueta o intenta configurar valores para variables de etiquetas no existentes.

- **Ignorar exceso de contenido de variables:** trunca los valores de datos que exceden la longitud de la variable tal lo definido en el diseñador de etiquetas para que se ajusten. Esta opción está en efecto cuando se configuran valores de variables en los filtros, desde los archivos de comandos y cuando se configuran valores de variables de desencadenadores para las variables de etiquetas del mismo nombre.

### Ejemplo

La variable de la etiqueta acepta 5 caracteres como máximo. Con esta opción habilitada, cualquier valor de más de cinco caracteres se trunca a los primeros cinco caracteres. Si el valor es 1234567 ignora los dígitos 6 y 7.

- **Ignorar variables de etiquetas faltantes:** Al imprimir con [archivos de comandos](#) (como un archivo JOB), el proceso de impresión ignora todas las variables que están:
  - especificadas en el archivo de comandos (que usan el comando [SET](#))
  - no definidas en la etiqueta

Lo mismo sucede si define el área de asignación en un filtro para que extraiga todos los pares nombre-valor, pero su etiqueta contiene menos variables.

Al configurar valores de variables de etiquetas no existentes, informo un error. Si esta opción está habilitada la impresión continúa.

Las opciones en la sección **Secuencia de comandos** especifican las posibilidades de las secuencias de comandos.

- **Idioma de secuencias:** Selecciona el idioma de secuencias para el desencadenador. Todas las acciones **Ejecutar secuencia de comandos** que utiliza dentro de un solo desencadenador usan el idioma de secuencia de comandos. seleccionado.

Las opciones en la sección **Guardar datos recibidos** especifican los comandos disponibles para los datos recibidos por el desencadenador.

- **Guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción y guarde los datos recibidos por el desencadenador. La opción **Variable** habilita un nombre de archivo variable. Seleccione una variable que contenga la ruta y el nombre de archivo.
- **Al guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción para guardar los datos en el desencadenador sólo si se produce un error durante la ejecución de la acción. Puede activar esta opción para mantener los datos que causaron el problema listos para resolver el problema más adelante.



#### NOTA

Asegúrese de habilitar el soporte para impresión supervisada. De lo contrario, no puede detectar errores durante la ejecución. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).



#### NOTA

guarda los datos recibidos en un archivo temporal. Este archivo temporal se elimina inmediatamente después de que se completa la ejecución del desencadenador. La variable interna `DataFileName` hace referencia a ese nombre de archivo. Para obtener más información, consulte el tema [Variables internas](#).

### Seguridad

- **Bloquear y cifrar desencadenador:** Habilita la protección del desencadenador. Si lo habilita, el desencadenador queda bloqueado y usted ya no puede editarlo. Esto cifra las acciones. Únicamente el usuario que tenga una contraseña puede desbloquear el desencadenador y modificarlo.

## 3.4. Desencadenador de servidor TCP/IP

Para saber más sobre los desencadenadores en general, consulte el tema [Comprender los desencadenadores](#).

El evento de desencadenador de servidor TCP/IP se produce cuando se reciben datos en el socket monitoreado (dirección IP y número de puerto).

Uso típico: El sistema de negocio existente ejecuta una transacción, que en efecto envía los datos al servidor NiceLabel Automation en un socket específico. El contenido de los datos puede estructurarse en CSV, XML y otros formatos, o bien en un formato heredado. De cualquier modo, NiceLabel Automation leerá los datos, analizará los valores utilizando los filtros y los imprimirá en etiquetas. Para obtener más información sobre el modo de analizar y extraer datos, consulte la sección [Comprender los filtros](#).

### General



#### NOTA

Este desencadenador admite el Protocolo de Internet versión 6 (IPv6).

Esta sección le permite configurar los parámetros más importantes del desencadenador de archivo.

- **Nombre:** Especifica el nombre único del desencadenador. El nombre lo ayuda a distinguir los diferentes desencadenadores cuando los configura en Automation Builder y luego cuando los ejecuta en Automation Manager.
- **Descripción:** Le permite describir el rol de este desencadenador. Ingrese una breve explicación acerca de lo que hace el desencadenador para ayudar a los usuarios.
- **Puerto:** Especifica el número de puerto donde se aceptarán los datos de entrada. Utilice el número de puerto que no está en uso por otra aplicación. Si el puerto seleccionado está en uso, no podrá iniciar el desencadenador en Automation Manager. Para obtener más información sobre problemas de seguridad, consulte la sección [Asegurar el acceso a sus desencadenadores](#).



#### NOTA

Si su servidor tiene hospedaje múltiple habilitado (más direcciones IP en una o más tarjetas de red), NiceLabel Automation responderá en el puerto definido en todas las direcciones IP.

- **Número máximo de llamadas concurrentes:** especifica el número máximo de conexiones aceptadas. Esa cantidad de clientes simultáneos pueden enviar datos al desencadenador al mismo tiempo.

Las opciones de la sección **Evento de ejecución** especifican cuándo se debe disparar el desencadenador y comenzar a ejecutar las acciones.

- **Cuando se desconecta el cliente:** Especifica que el desencadenador se disparará después de que el cliente envíe datos y cierre la conexión. Este es un parámetro predeterminado.



#### NOTA

Si desea enviar el estado del trabajo de impresión a la aplicación de terceros como comentarios, no utilice esta opción. Si la conexión se deja abierta, pueden enviarse comentarios utilizando la acción **Enviar datos a puerto TCP/IP** con el parámetro *Responder al remitente*.

- **Cuando se recibe cierta cantidad de caracteres:** especifica que el desencadenador se dispara cada vez que se recibe la cantidad requerida de caracteres. En este caso, la aplicación de terceros puede mantener una conexión abierta y enviar datos continuamente. Todos los segmentos de datos deben tener el mismo tamaño.
- **Cuando se recibe una secuencia de caracteres:** Especifica que el desencadenador se dispara cada vez que se recibe la secuencia requerida de caracteres. Utilice esta opción si sabe que el "final de los datos" siempre está identificado por un conjunto único de caracteres. Puede insertar caracteres especiales (binarios) utilizando el botón que se encuentra junto al campo de edición.
  - **Incluir en datos de desencadenador:** La secuencia de caracteres que determina que el evento de desencadenador no se elimina de los datos, sino que se incluye junto con los datos. El evento recibe la secuencia completa de datos recibidos.



- **Cuando no se recibe nada después de un intervalo específico de tiempo:** especifica que el desencadenador se activa después de que transcurre un intervalo específico de tiempo después de que se recibe el último carácter.

## Ejecución

- **Permitir conexiones desde los siguientes hosts:** Especifica la lista de direcciones IP o nombres de host de computadoras que tienen permitido conectarse al desencadenador. Coloque cada entrada en una nueva línea.
- **Negar conexiones desde los siguientes hosts:** Especifica la lista de direcciones IP o nombres de host de computadoras que no tienen permitido conectarse al desencadenador. Coloque cada entrada en una nueva línea.
- **Mensaje de bienvenida:** Especifica el mensaje de texto que se muestra al cliente cada vez que se conecta al desencadenador TCP/IP.
- **Mensaje de respuesta:** Especifica el mensaje de texto que se muestra al cliente cada vez que se ejecutan acciones. Utilice esta opción cuando el cliente no se desconecta después de enviar datos y espera la respuesta cuando termina la ejecución de acciones. El mensaje de respuesta tiene una codificación rígida y, por lo tanto, es siempre el mismo.
- **Codificación de mensaje:** Especifica el esquema de codificación de datos para que los caracteres especiales puedan procesarse correctamente. NiceLabel Automation puede detectar automáticamente la codificación de datos, en función del encabezado BOM (archivos de texto) o del atributo de codificación (archivos XML).

## Otros

Las opciones en la sección **Comentarios del motor de impresión** especifican los parámetros de comunicación que le permiten recibir comentarios del motor de impresión.

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✘	✔	✔

- **Impresión supervisada:** Activa el modo de impresión sincronizada. Utilícela siempre que quiera enviar el estado del trabajo de impresión a una aplicación de terceros. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

Las opciones en la sección **Procesamiento de datos** especifican si quiere recortar los datos para que entren en la variable o ignorar las variables de las etiquetas faltantes. En forma predeterminada, notificará los errores e interrumpirá el proceso de impresión si intenta guardar valores muy largos en la variable de la etiqueta o intenta configurar valores para variables de etiquetas no existentes.

- **Ignorar exceso de contenido de variables:** trunca los valores de datos que exceden la longitud de la variable tal lo definido en el diseñador de etiquetas para que se ajusten. Esta opción está en efecto cuando se configuran valores de variables en los filtros, desde los archivos de comandos y cuando se configuran valores de variables de desencadenadores para las variables de etiquetas del mismo nombre.

## Ejemplo

La variable de la etiqueta acepta 5 caracteres como máximo. Con esta opción habilitada, cualquier valor de más de cinco caracteres se trunca a los primeros cinco caracteres. Si el valor es 1234567 ignora los dígitos 6 y 7.

- **Ignorar variables de etiquetas faltantes:** Al imprimir con [archivos de comandos](#) (como un archivo JOB), el proceso de impresión ignora todas las variables que están:
  - especificadas en el archivo de comandos (que usan el comando [SET](#))
  - no definidas en la etiqueta

Lo mismo sucede si define el área de asignación en un filtro para que extraiga todos los pares nombre-valor, pero su etiqueta contiene menos variables.

Al configurar valores de variables de etiquetas no existentes, informo un error. Si esta opción está habilitada la impresión continúa.

Las opciones en la sección **Secuencia de comandos** especifican las posibilidades de las secuencias de comandos.

- **Idioma de secuencias:** Selecciona el idioma de secuencias para el desencadenador. Todas las acciones **Ejecutar secuencia de comandos** que utiliza dentro de un solo desencadenador usan el idioma de secuencia de comandos. seleccionado.

Las opciones en la sección **Guardar datos recibidos** especifican los comandos disponibles para los datos recibidos por el desencadenador.

- **Guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción y guarde los datos recibidos por el desencadenador. La opción **Variable** habilita un nombre de archivo variable. Seleccione una variable que contenga la ruta y el nombre de archivo.
- **Al guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción para guardar los datos en el desencadenador sólo si se produce un error durante la ejecución de la acción. Puede activar esta opción para mantener los datos que causaron el problema listos para resolver el problema más adelante.



### NOTA

Asegúrese de habilitar el soporte para impresión supervisada. De lo contrario, Automation no puede detectar errores durante la ejecución. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).



### NOTA

Automation guarda los datos recibidos en un archivo temporal. Este archivo temporal se elimina inmediatamente después de que se completa la ejecución del desencadenador. La variable interna `DataFileName` hace referencia a ese nombre de archivo. Para obtener más información, consulte el tema [Variables internas](#).

## Seguridad

- **Bloquear y cifrar desencadenador:** Habilita la protección del desencadenador. Si lo habilita, el desencadenador queda bloqueado y usted ya no puede editarlo. Esto cifra las acciones. Únicamente el usuario que tenga una contraseña puede desbloquear el desencadenador y modificarlo.

## Enviar señales keepalive

El sistema de enrutamiento de la red puede perder la conexión sin avisar cuando no hay tráfico TCP/IP durante algunos minutos. A fin de evitar la desconexión, puede hacer que se envíen periódicamente señales keepalive en NiceLabel Automation.

Habilite el envío de señales keepalive en su archivo `product.config`:

1. Navegue a la carpeta Sistema.  
%PROGRAMDATA%\NiceLabel\NiceLabel 10
2. Haga una copia de seguridad del archivo `product.config`.
3. Abra `product.config` en el editor de texto. El archivo tiene una estructura XML.
4. Agregue las siguientes líneas:

```
<configuration>
  <IntegrationService>
    <KeepAliveTime>60000</KeepAliveTime>
    <KeepAliveInterval>10000</KeepAliveInterval>
  </IntegrationService>
</configuration>
```



### NOTA

`KeepAliveTime` (en milisegundos): Especifica durante cuánto tiempo debe estar inactiva la conexión del socket de conexión TCP antes de que Automation envíe una señal keepalive y espere a que se responda con paquetes de reconocimiento.

`KeepAliveInterval` (en milisegundos): Especifica los intervalos entre el envío de un paquete keepalive y otro en caso de que el host no responda con ningún paquete de reconocimiento.

Cuando el desencadenador TCP/IP recibe el paquete keepalive y responde con un paquete de reconocimiento, el temporizador de `KeepAliveTime` se reinicia.

**Ambos valores (`KeepAliveTime` y `KeepAliveInterval`) son obligatorios para habilitar el envío de señales keepalive, y deben tener un valor superior a 0.**

5. Guarde el archivo `product.config`.
6. Reinicie su Automation service.

Una vez habilitadas las señales keepalive, la conexión se mantendrá activa.

## 3.5. Desencadenador de cliente TCP/IP

EDITION AVAILABILITY:	Automation Easy ✗	Automation Pro ✓	Automation Enterprise ✓
-----------------------	----------------------	---------------------	----------------------------

Para saber más sobre los desencadenadores en general, consulte el tema [Comprender los desencadenadores](#).

El desencadenador de cliente TCP/IP convierte su Automation en un cliente de escucha que se conecta a servidores TCP/IP. Existen diversos dispositivos y sistemas que desempeñan la función de un servidor TCP/IP: sistemas de inspección visual, impresoras, PLC, escáneres, balanzas, etc. Automation puede conectarse a ellos y esperar los datos entrantes. Después de recibir una cierta cantidad de caracteres, una secuencia de caracteres o un tiempo de espera, el desencadenador de cliente TCP/IP se activa y comienza a ejecutar sus acciones. Si la conexión falla, el desencadenador le permite volver a conectarse automáticamente.

Uso típico: Está imprimiendo de forma automática varios tipos de etiquetas de embalaje con una impresora de red. Es fundamental que sepa cuándo la impresora ha terminado de imprimir un tipo de etiqueta antes de comenzar a imprimir el segundo tipo de etiqueta. El desencadenador de cliente TCP/IP le permite establecer la verificación automática permanente de la disponibilidad de la impresora. Mientras imprime el primer tipo de etiqueta, el desencadenador de cliente TCP/IP verifica el estado de impresión y, cuando la impresora está lista, envía un mensaje a Automation para enviar el segundo tipo de etiqueta para su impresión.

### General

- **Nombre:** Especifica el nombre único del desencadenador. El nombre lo ayuda a distinguir los diferentes desencadenadores cuando los configura en Automation Builder y luego cuando los ejecuta en Automation Manager.
- **Descripción:** Le permite describir el rol de este desencadenador. Ingrese una breve explicación acerca de lo que hace el desencadenador para ayudar a los usuarios.
- **Servidor de destino:** Escriba la ubicación (dirección IP o nombre de host) del servidor TCP/IP al que desea conectarse.
- **Puerto:** Especifica el número de puerto del host desde el que recibirá los datos entrantes. Asegúrese de que los puertos del firewall estén abiertos del lado del servidor.
- **Reconectarse al intervalo del servidor:** Defina el tiempo en milisegundos después del cual Automation intenta volver a conectarse a su servidor TCP/IP.
- **Cuando se recibe cierta cantidad de caracteres:** especifica que el desencadenador se activa cada vez que se recibe la cantidad requerida de caracteres. En este caso, el servidor puede mantener una conexión abierta y enviar datos continuamente. Todos los segmentos de datos deben tener el mismo tamaño.
- **Cuando se recibe una secuencia de caracteres:** Especifica que el desencadenador se dispara cada vez que se recibe la secuencia requerida de caracteres. Utilice esta opción si sabe que el "final de

los datos" siempre está identificado por un conjunto único de caracteres. Puede insertar caracteres especiales (binarios) utilizando el botón que se encuentra junto al campo de edición.

- **Incluir en datos de desencadenador:** La secuencia de caracteres que determina que el evento de desencadenador no se elimina de los datos, sino que se incluye junto con los datos. El evento recibe la secuencia completa de datos recibidos.
- **Si no se recibe nada después de un intervalo específico de tiempo:** especifica que el desencadenador se activa después de que transcurre un intervalo específico de tiempo después de que se recibe el último carácter.

## Ejecución

- **Mensaje de inicialización:** mensaje de texto que se envía al servidor cada vez que Automation establece conexión.
- **Mensaje de respuesta:** Especifica el mensaje de texto que vuelve al servidor al activar el desencadenador (antes de que las acciones comiencen a ejecutarse).
- **Codificación de mensaje:** Especifica el esquema de codificación de datos para que los caracteres especiales puedan procesarse correctamente. NiceLabel Automation puede detectar automáticamente la codificación de datos, en función del encabezado BOM (archivos de texto) o del atributo de codificación (archivos XML).

## Otros

Las opciones en la sección **Comentarios del motor de impresión** especifican los parámetros de comunicación que le permiten recibir comentarios del motor de impresión.

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✗	✓	✓

- **Impresión supervisada:** Activa el modo de impresión sincronizada. Utilícela siempre que quiera enviar el estado del trabajo de impresión a una aplicación de terceros. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

Las opciones en la sección **Procesamiento de datos** especifican si quiere recortar los datos para que entren en la variable o ignorar las variables de las etiquetas faltantes. En forma predeterminada, notificará los errores e interrumpirá el proceso de impresión si intenta guardar valores muy largos en la variable de la etiqueta o intenta configurar valores para variables de etiquetas no existentes.

- **Ignorar exceso de contenido de variables:** trunca los valores de datos que exceden la longitud de la variable tal lo definido en el diseñador de etiquetas para que se ajusten. Esta opción está en efecto cuando se configuran valores de variables en los filtros, desde los archivos de comandos y cuando se configuran valores de variables de desencadenadores para las variables de etiquetas del mismo nombre.

## Ejemplo

La variable de la etiqueta acepta 5 caracteres como máximo. Con esta opción habilitada, cualquier valor de más de cinco caracteres se trunca a los primeros cinco caracteres. Si el valor es 1234567 ignora los dígitos 6 y 7.

- **Ignorar variables de etiquetas faltantes:** Al imprimir con [archivos de comandos](#) (como un archivo JOB), el proceso de impresión ignora todas las variables que están:
  - especificadas en el archivo de comandos (que usan el comando [SET](#))
  - no definidas en la etiqueta

Lo mismo sucede si define el área de asignación en un filtro para que extraiga todos los pares nombre-valor, pero su etiqueta contiene menos variables.

Al configurar valores de variables de etiquetas no existentes, informo un error. Si esta opción está habilitada la impresión continúa.

Las opciones en la sección **Secuencia de comandos** especifican las posibilidades de las secuencias de comandos.

- **Idioma de secuencias:** Selecciona el idioma de secuencias para el desencadenador. Todas las acciones **Ejecutar secuencia de comandos** que utiliza dentro de un solo desencadenador usan el idioma de secuencia de comandos. seleccionado.

Las opciones en la sección **Guardar datos recibidos** especifican los comandos disponibles para los datos recibidos por el desencadenador.

- **Guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción y guarde los datos recibidos por el desencadenador. La opción **Variable** habilita un nombre de archivo variable. Seleccione una variable que contenga la ruta y el nombre de archivo.
- **Al guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción para guardar los datos en el desencadenador sólo si se produce un error durante la ejecución de la acción. Puede activar esta opción para mantener los datos que causaron el problema listos para resolver el problema más adelante.



### NOTA

Asegúrese de habilitar el soporte para impresión supervisada. De lo contrario, no puede detectar errores durante la ejecución. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).



### NOTA

guarda los datos recibidos en un archivo temporal. Este archivo temporal se elimina inmediatamente después de que se completa la ejecución del desencadenador. La variable interna `DataFileName` hace referencia a ese nombre de archivo. Para obtener más información, consulte el tema [Variables internas](#).

## Seguridad

- **Bloquear y cifrar desencadenador:** Habilita la protección del desencadenador. Si lo habilita, el desencadenador queda bloqueado y usted ya no puede editarlo. Esto cifra las acciones. Únicamente el usuario que tenga una contraseña puede desbloquear el desencadenador y modificarlo.

## Enviar señales keepalive

El sistema de enrutamiento de la red puede perder la conexión sin avisar cuando no hay tráfico TCP/IP durante algunos minutos. A fin de evitar la desconexión, puede hacer que se envíen periódicamente señales keepalive en NiceLabel Automation.

Habilite el envío de señales keepalive en su archivo `product.config`:

1. Navegue a la carpeta Sistema.  
%PROGRAMDATA%\NiceLabel\NiceLabel 10
2. Haga una copia de seguridad del archivo `product.config`.
3. Abra `product.config` en el editor de texto. El archivo tiene una estructura XML.
4. Agregue las siguientes líneas:

```
<configuration>
  <IntegrationService>
    <KeepAliveTime>60000</KeepAliveTime>
    <KeepAliveInterval>10000</KeepAliveInterval>
  </IntegrationService>
</configuration>
```



### NOTA

`KeepAliveTime` (en milisegundos): Especifica durante cuánto tiempo debe estar inactiva la conexión del socket de conexión TCP antes de que Automation envíe una señal keepalive y espere a que se responda con paquetes de reconocimiento.

`KeepAliveInterval` (en milisegundos): Especifica los intervalos entre el envío de un paquete keepalive y otro en caso de que el host no responda con ningún paquete de reconocimiento.

Cuando el desencadenador TCP/IP recibe el paquete keepalive y responde con un paquete de reconocimiento, el temporizador de `KeepAliveTime` se reinicia.

**Ambos valores (`KeepAliveTime` y `KeepAliveInterval`) son obligatorios para habilitar el envío de señales keepalive, y deben tener un valor superior a 0.**

5. Guarde el archivo `product.config`.
6. Reinicie su Automation service.

Una vez habilitadas las señales keepalive, la conexión se mantendrá activa.

## 3.6. Desencadenador de servidor HTTP

EDITION AVAILABILITY:	Automation Easy ✗	Automation Pro ✓	Automation Enterprise ✓
-----------------------	----------------------	---------------------	----------------------------

Para saber más sobre los desencadenadores en general, consulte el tema [Comprender los desencadenadores](#).

El evento de desencadenador HTTP se dispara si se reciben datos en el socket monitoreado (dirección IP y número de puerto).

A diferencia de lo que sucede con el desencadenador TCP/IP, los datos recibidos no están en una secuencia de datos sin procesar, sino que incluyen el encabezado HTTP estándar. Las aplicaciones de terceros deben usar los métodos de solicitud POST o GET. Deben proporcionar datos en el cuerpo del mensaje o en una cadena de búsqueda. Puede usar ambos tipos de medios de Internet en el cuerpo de mensaje: tipo MIME o tipo de contenido. NiceLabel Automation recibe el mensaje y extraer los datos relevantes del contenido del mensaje usando un filtro.

Uso típico: el sistema de negocio existente ejecuta una transacción, que envía los datos al servidor NiceLabel Automation con formato de mensaje HTTP POST en un socket específico. Los datos enviados pueden estar estructurados en CSV, XML y otros formatos, o bien pueden estructurarse usando formato heredado propio. De cualquier modo, NiceLabel Automation leerá los datos, analizará los valores utilizando los filtros y los imprimirá los datos extraídos en etiquetas. Para obtener más información sobre el modo de analizar y extraer datos, consulte la sección [Comprender los filtros](#).

### Suministrar datos

Proporcione datos al desencadenador HTTP usando cualquiera de los siguientes métodos. De ser necesario, también puede combinar los métodos y utilizar ambos en la misma solicitud HTTP.



### Datos en la cadena de consulta

Una cadena de consulta es la parte de un localizador uniforme de recursos (URL) que contiene los datos que se pasarán al desencadenador HTTP.

Un ejemplo de un URL típico que contiene una cadena de consulta:

```
http://server/path/?query_string
```

El signo de pregunta se utiliza como separador y no forma parte de la cadena de consulta.

La cadena de consulta generalmente está compuesta de una serie de pares **name:value**. Dentro de cada par, el nombre de campo y el valor están separados por signos igual (=). La serie de pares se separa mediante la Y comercial (&). Una cadena de consulta típica proporciona valores para campos (variables) en el siguiente formato:

```
field1=value1&field2=value2&field3=value3
```

El desencadenador HTTP ofrece soporte integrado para extraer valores de todos los campos y almacenarlos en variables que llevan el mismo nombre. Como resultado, no debe definir ningún filtro para extraer valores de la cadena de consulta.

- No debe definir variables dentro de un desencadenador para cargarlas con valores de la cadena de consulta. NiceLabel Automation extraerá todas las variables en la cadena de consulta y enviará los valores a la etiqueta activa. Si existen variables con el mismo nombre en la etiqueta, Automation los carga con valores. Si las variables no existen en la etiqueta, Automation ignora sus valores sin informar ningún error.
- Si una acción requiere valores de variables, defina las variables correspondientes en el desencadenador. Para obtener y almacenar todos los valores de la cadena de consulta, cree variables correspondientes usando los mismos nombres como campos de la cadena de consulta. Para el ejemplo anterior, tiene que definir las variables del desencadenador con los nombres **field1**, **field2** y **field3**.

Por lo general, debe utilizar el método de solicitud GET HTTP para proporcionar la cadena de consulta.

### Datos en el cuerpo de la solicitud HTTP

Debe utilizar el método de solicitud para proporcionar el mensaje en el cuerpo de la solicitud HTTP.

Puede enviar cualquier tipo de datos o usar cualquier estructura de datos que desee en el cuerpo. Solo debe asegurarse de poder manejar los datos usando NiceLabel Automation filtros. El contenido puede tener formato como XML, CSV o texto sin formato. El contenido enviado puede ser incluso datos binarios (codificado en Base64).. Tenga en cuenta que deberá analizar los datos con filtros.

Si puede influenciar en la estructura del mensaje de entrada, utilice estructuras estandarizadas, como XML o CSV, para simplificar la configuración de los filtros.

Utilice el método de solicitud POST HTTP para proporcionar los datos en el cuerpo del mensaje.

## General

Esta sección le permite configurar los parámetros más importantes del desencadenador de archivo.

- **Nombre:** Especifica el nombre único del desencadenador. El nombre lo ayuda a distinguir los diferentes desencadenadores cuando los configura en Automation Builder y luego cuando los ejecuta en Automation Manager.
- **Descripción:** Le permite describir el rol de este desencadenador. Ingrese una breve explicación acerca de lo que hace el desencadenador para ayudar a los usuarios.

## Comunicación



### NOTA

Este desencadenador admite el Protocolo de Internet versión 6 (IPv6).

Esta sección le permite configurar el número de puerto obligatorio y las opciones de comentarios opcionales. Puede utilizar los Códigos de respuesta HTTP estándares para indicar que la acción se ejecutó correctamente. Para propósitos más avanzados, también puede enviar el contenido personalizado a la aplicación que suministra los datos. Dicho contenido puede ser simple cadena de comentarios o datos binarios, tales como vista previa de etiqueta o secuencia de impresión.

El URL típico para conectar al desencadenador HTTP es el siguiente:

```
http://server:port/path/?query_string
```

- **Servidor:** Es la dirección IP o FQDN de la máquina en la que se instaló NiceLabel Automation.
- **Puerto:** Número del puerto en el que se reciben los datos entrantes. Utilice el número de puerto que no está en uso por otra aplicación. Si el puerto seleccionado está en uso, no podrá iniciar el desencadenador en Automation Manager. Para obtener más información sobre problemas de seguridad, consulte la sección [Asegurar el acceso a sus desencadenadores](#).



### NOTA

Si su servidor tiene hospedaje múltiple habilitado (múltiples direcciones IP en una o más tarjetas de red), NiceLabel Automation responderá en el puerto definido en todas las direcciones IP.

- **Ruta:** Especifica la ruta de acceso opcional en el URL. Esta funcionalidad permite a NiceLabel Automation exponer múltiples desencadenadores HTTP en el mismo puerto. El cliente utilizará los desencadenadores a través de un solo puerto en una sintaxis tipo REST, lo que ocasiona que los distintos desencadenadores se disparen por un URL diferente. Si no está seguro sobre qué utilizar, deje la ruta de acceso predeterminada (\).

<b>EDITION AVAILABILITY:</b>	Automation Easy ✗	Automation Pro ✗	Automation Enterprise ✓
------------------------------	----------------------	---------------------	----------------------------

- **Conexión segura (HTTPS):** habilita una capa de transporte seguro para su mensaje HTTP y previene la interceptación de usuarios no autorizados. Para obtener más información sobre el modo de asegurar la conexión, consulte la sección [Usar capa de transporte seguro \(HTTPS\)](#).
- **Cadena de consulta:** Especifica los pares nombre-valor en el URL. Este es un parámetro opcional. Los datos se proporcionan por lo general en el cuerpo de la solicitud HTTP.
- **Esperar a que termine la ejecución del desencadenador:** El protocolo HTTP requiere que el receptor (en este caso NiceLabel Automation) envíe una respuesta numérica al remitente para indicar el estado del mensaje recibido. De manera predeterminada, NiceLabel Automation responde con el código 200. Esto indica que Automation recibió correctamente los datos, pero no da información acerca del éxito de las acciones del desencadenador.

Esta opción especifica que un desencadenador no envía una respuesta inmediatamente después de recibir los datos, sino que espera hasta que la acción se ejecute. Luego envía el código de respuesta indicando una ejecución de la acción exitosa. Cuando esta opción está habilitada, puede enviar un tipo de respuesta y datos personalizados (por ejemplo, la respuesta a una solicitud HTTP es una vista previa de etiqueta en formato PDF).

Los códigos de respuesta HTTP integrados disponibles son:

Código de respuesta HTTP	Descripción
200	Todas las acciones ejecutadas correctamente.
401	No autorizado, se especificó un nombre de usuario y una contraseña incorrectos.
500	Se produjeron errores durante la ejecución de la acción.



#### NOTA

Para enviar comentarios acerca del proceso de impresión, habilite el modo de impresión **sincrónico**. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

- **Número máximo de llamadas simultáneas:** especifica el número máximo de conexiones entrantes simultáneas. Esa cantidad de clientes simultáneos pueden enviar datos al desencadenador al mismo tiempo. La cantidad también depende del rendimiento de hardware de su servidor. Obtenga más información sobre [Sección 7.1, "Procesamiento paralelo"](#).
- **Tipo de respuesta:** Especifica el tipo de mensaje de respuesta. Los tipos de medios de Internet utilizados frecuentemente (también denominados "tipos MIME" o "tipos de contenido") se predefinen en la casilla desplegable. Si su tipo de medio no está disponible en la lista, ingréselo usted mismo. Automation envía los datos de respuestas de salida como comentario, con formato en el tipo de medio definido. **Variable** habilita los tipos de medio variables. Si se habilita, seleccione o cree una variable que contenga el tipo de medio.



## NOTA

Si no especifica el tipo de contenido, NiceLabel Automation utilizará `application/octet-stream` como opción predeterminada.

- **Datos de respuesta:** Especifica el contenido de su mensaje de respuesta. Algunos ejemplos de lo que puede enviar como una respuesta HTTP: mensajes de error personalizados, vista previa de etiqueta, archivos PDF generados, archivo de secuencia de impresión (archivo de cola de impresión), archivo XML con detalles del motor de impresión más la vista previa de etiqueta (codificada como cadena Base64). Las posibilidades son ilimitadas.

Si enviará contenido únicamente binario (como una vista previa de etiqueta o secuencia de impresión), asegúrese de seleccionar el tipo de medio adecuado; por ejemplo, `image/jpeg` o `application/octet-stream`.

- **Encabezados adicionales:** Le permiten definir encabezados personalizados tipo MIME para el mensaje de respuesta HTTP.

Encontrará la sintaxis del encabezado de la respuesta y ejemplos en la [sección acción Solicitud HTTP](#).



## SUGERENCIA

Con los datos de la respuesta y los Encabezados adicionales, puede utilizar contenido fijo, una combinación de contenido fijo y variable, o contenido únicamente variable. Para insertar contenido variable, haga clic en el botón con flecha que se encuentra a la derecha del área de datos e inserte la variable desde la lista (o cree una nueva) que contiene los datos que desea usar. Para obtener más información, consulte la sección [Usar valores compuestos](#).

## Autenticación

- **Ninguno:** No hay ningún método de autenticación en uso.
- **Usuario:** Especifica que los mensajes de entrada incluyen nombre de usuario y contraseña. Cuando está en uso, el desencadenador solamente acepta mensajes HTTP cuyas credenciales coincidan. Para obtener más información sobre problemas de seguridad, consulte la sección [Asegurar el acceso a sus desencadenadores](#).
- **Grupo de aplicación (definido NiceLabel Control Center):** Como en el caso del tipo de autenticación **Usuario**, esta opción también especifica que los mensajes de entrada incluyen nombre de usuario y contraseña. Cuando está en uso el desencadenador solo acepta mensajes HTTP con las credenciales adecuadas para usuarios NiceLabel Control Center que pertenecen a un grupo de aplicación específico.
  - **Grupo:** Pueden definirse múltiples grupos de la aplicación en el NiceLabel Control Center. Para ganar que el grupo debe tener acceso al desencadenador del servidor HTTP, use la lista desplegable **Grupo**. El grupo seleccionado y sus usuarios deben configurarse como activos cuando el desencadenador se está ejecutando.



## NOTA

El grupo con un nombre específico debe existir en NiceLabel Control Center cuando el desencadenador se está ejecutando. Mientras trabaja en la configuración en Automation Builder, puede utilizar cualquier nombre de grupo. Asegúrese de eventualmente definir un nombre final en NiceLabel Control Center y hágalo coincidir en la configuración antes de implementarlo.






## SUGERENCIA

Los usuarios se autentican a sí mismos usando sus credenciales según lo definido en **NiceLabel Control Center**, pestaña **Usuarios**. Consulte la Guía del usuario NiceLabel Control Center para obtener más detalles sobre la administración de usuarios (sección Usuarios y grupos).

### Otros

Las opciones en la sección **Comentarios del motor de impresión** especifican los parámetros de comunicación que le permiten recibir comentarios del motor de impresión.

<b>EDITION AVAILABILITY:</b>	Automation Easy 	Automation Pro 	Automation Enterprise 
------------------------------	--	---	--

- **Impresión supervisada:** Activa el modo de impresión sincronizada. Utilícela siempre que quiera enviar el estado del trabajo de impresión a una aplicación de terceros. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

Las opciones en la sección **Procesamiento de datos** especifican si quiere recortar los datos para que entren en la variable o ignorar las variables de las etiquetas faltantes. En forma predeterminada, notificará los errores e interrumpirá el proceso de impresión si intenta guardar valores muy largos en la variable de la etiqueta o intenta configurar valores para variables de etiquetas no existentes.

- **Ignorar exceso de contenido de variables:** trunca los valores de datos que exceden la longitud de la variable tal lo definido en el diseñador de etiquetas para que se ajusten. Esta opción está en efecto cuando se configuran valores de variables en los filtros, desde los archivos de comandos y cuando se configuran valores de variables de desencadenadores para las variables de etiquetas del mismo nombre.

### Ejemplo

La variable de la etiqueta acepta 5 caracteres como máximo. Con esta opción habilitada, cualquier valor de más de cinco caracteres se trunca a los primeros cinco caracteres. Si el valor es 1234567 ignora los dígitos 6 y 7.

- **Ignorar variables de etiquetas faltantes:** Al imprimir con [archivos de comandos](#) (como un archivo JOB), el proceso de impresión ignora todas las variables que están:

- especificadas en el archivo de comandos (que usan el comando [SET](#))
- no definidas en la etiqueta

Lo mismo sucede si define el área de asignación en un filtro para que extraiga todos los pares nombre-valor, pero su etiqueta contiene menos variables.

Al configurar valores de variables de etiquetas no existentes, informo un error. Si esta opción está habilitada la impresión continúa.

Las opciones en la sección **Secuencia de comandos** especifican las posibilidades de las secuencias de comandos.

- **Idioma de secuencias:** Selecciona el idioma de secuencias para el desencadenador. Todas las acciones **Ejecutar secuencia de comandos** que utiliza dentro de un solo desencadenador usan el idioma de secuencia de comandos. seleccionado.

Las opciones en la sección **Guardar datos recibidos** especifican los comandos disponibles para los datos recibidos por el desencadenador.

- **Guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción y guarde los datos recibidos por el desencadenador. La opción **Variable** habilita un nombre de archivo variable. Seleccione una variable que contenga la ruta y el nombre de archivo.
- **Al guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción para guardar los datos en el desencadenador sólo si se produce un error durante la ejecución de la acción. Puede activar esta opción para mantener los datos que causaron el problema listos para resolver el problema más adelante.



#### NOTA

Asegúrese de habilitar el soporte para impresión supervisada. De lo contrario, no puede detectar errores durante la ejecución. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).



#### NOTA

guarda los datos recibidos en un archivo temporal. Este archivo temporal se elimina inmediatamente después de que se completa la ejecución del desencadenador. La variable interna `DataFileName` hace referencia a ese nombre de archivo. Para obtener más información, consulte el tema [Variables internas](#).

## Seguridad

- **Bloquear y cifrar desencadenador:** Habilita la protección del desencadenador. Si lo habilita, el desencadenador queda bloqueado y usted ya no puede editarlo. Esto cifra las acciones. Únicamente el usuario que tenga una contraseña puede desbloquear el desencadenador y modificarlo.

## 3.7. Desencadenador de servicio Web

EDITION AVAILABILITY:	Automation Easy ✗	Automation Pro ✗	Automation Enterprise ✓
-----------------------	----------------------	---------------------	----------------------------

Para saber más sobre los desencadenadores en general, consulte el tema [Comprender los desencadenadores](#).

El evento de desencadenador de servicio Web se produce cuando se reciben datos en el socket monitoreado (dirección IP y número de puerto). Los datos deben seguir la notación SOAP - codifica datos XML en el mensaje HTTP. La interfaz del servicio Web se describe con el documento WSDL. Dicho documento está disponible cada desencadenador de servicio Web.

El desencadenador de servicio Web puede proporcionar comentarios sobre el estado del trabajo de impresión, pero debe habilitarse el modo de procesamiento **sincronizado**. Para obtener más información, consulte la sección [Comentarios sobre el estado del trabajo de impresión](#).

Generalmente, los programadores usan el servicio Web para integrar la impresión de etiquetas en sus propias aplicaciones. Un sistema empresarial existente ejecuta una transacción, que envía los datos al servidor NiceLabel Automation en un socket específico. Los datos enviados se envían con formato de mensaje SOAP. Los datos pueden proporcionarse en CSV, XML y otros formatos estructurados, o bien en algún formato heredado. De cualquier modo, NiceLabel Automation leerá los datos, analizará los valores utilizando los filtros y los imprimirá en etiquetas. Para obtener más información sobre el modo de analizar y extraer datos, consulte la sección [Comprender los filtros](#).

### General

Esta sección le permite configurar los parámetros generales del desencadenador de archivo.

- **Nombre:** Especifica el nombre único del desencadenador. El nombre lo ayuda a distinguir los diferentes desencadenadores cuando los configura en Automation Builder y luego cuando los ejecuta en Automation Manager.
- **Descripción:** Le permite describir el rol de este desencadenador. Ingrese una breve explicación acerca de lo que hace el desencadenador para ayudar a los usuarios.

### Comunicación



#### NOTA

Este desencadenador admite el Protocolo de Internet versión 6 (IPv6).

Esta sección le permite configurar el número de puerto obligatorio y las opciones de comentarios opcionales.

- **Puerto:** especifica el número de puerto que acepta datos entrantes. Utilice el número de puerto que no está en uso por otra aplicación. Si el puerto seleccionado está en uso, no podrá iniciar

el desencadenador en Automation Manager. Para obtener más información sobre problemas de seguridad, consulte la sección [Asegurar el acceso a sus desencadenadores](#).



#### NOTA

Si su servidor tiene hospedaje múltiple habilitado (múltiples direcciones IP en una o más tarjetas de red), NiceLabel Automation responderá en el puerto definido en todas las direcciones IP.

- **Conexión segura (HTTPS):** habilita una capa de transporte seguro para su mensaje HTTP y previene la interceptación de usuarios no autorizados. Para obtener más información sobre el modo de asegurar la conexión, consulte la sección [Usar capa de transporte seguro \(HTTPS\)](#).
- **Número máximo de llamadas concurrentes:** especifica el número máximo de conexiones aceptadas. Esa cantidad de clientes simultáneos pueden enviar datos al desencadenador al mismo tiempo.
- **Datos de respuesta:** define la respuesta personalizada que puede utilizarse con los métodos `ExecuteTriggerWithResponse` y `ExecuteTriggerAndSetVariablesWithResponse`. El contenido de los datos de respuesta dependerá de lo que se indique en el área de texto. Puede combinar valores fijos, valores variables y caracteres especiales. Para insertar (o crear) variables y caracteres especiales, haga clic en el botón de flecha que se encuentra a la derecha del área de texto. La respuesta puede contener datos binarios, como una imagen de vista previa de etiqueta y un archivo de impresión (\*.PRN).

#### Comentarios sobre el estado

En su diseño, el desencadenador de servicio Web proporciona comentarios sobre el estado del trabajo de impresión creado. El desencadenador aceptará los datos proporcionados y los utilizará al ejecutar acciones definidas. Puede supervisar la ejecución de las acciones - el desencadenador informa estado correcto para cada evento que suceda durante la ejecución. Para habilitar informes de estado durante el proceso de impresión, active [Modo de impresión sincronizada](#).

El desencadenador de servicio Web expone los siguientes métodos (funciones):

- **ExecuteTrigger:** este método acepta los datos en el procesamiento y proporciona comentarios opcionales sobre el estado. Uno de los parámetros de entrada habilita o deshabilita los comentarios. Si habilita los informes de estado, los comentarios contendrán un ID y una descripción detallada del error. Si el ID de error es igual a 0, no hay ningún problema al crear el archivo de impresión. Si el ID de error es mayor que 0, se produjo algún error durante el proceso de impresión. La respuesta del servicio Web en este método no puede configurarse y contiene siempre el ID y la descripción del error.
- **ExecuteTriggerWithResponse:** este método acepta los datos en el procesamiento y proporciona comentarios personalizados sobre el estado. La respuesta del servicio Web puede configurarse. Puede responder usando cualquier tipo de datos organizados en cualquier estructura disponible. Puede utilizar datos binarios en la respuesta.
- **ExecuteTriggerAndSetVariables:** similar al descrito anteriormente, **ExecuteTrigger**, este servicio web expone un parámetro de entrada adicional que acepta la lista con formato de pares *nombre-valor*. El



desencadenador analizará automáticamente la lista, extraerá valores y los guardará en las variables del mismo nombre para que usted no tenga que crear ningún filtro de extracción de datos.

- **ExecuteTriggerAndSetVariablesWithResponse:** similar a **ExecuteTriggerWithResponse** descrito anteriormente, este servicio web expone un parámetro de entrada adicional que acepta la lista con formato de pares *nombre-valor*. El desencadenador analizará automáticamente la lista, extraerá valores y los guardará en las variables del mismo nombre para que usted no tenga que crear ningún filtro de extracción de datos.

Para obtener más información sobre la estructura de los mensajes que puede enviar a un método o el otro, consulte la sección [WSDL](#) más abajo.

## WSDL

El Lenguaje de descripción de servicios Web (WSDL) describe el estilo de los mensajes SOAP. Puede ser **Llamada de procedimiento remoto (RPC)** o un estilo de **documento**. Elija el estilo compatible con la aplicación que brinda los datos.

El documento WSDL define los parámetros de entrada y salida del servicio Web.

Una vez que define el desencadenador del servicio Web en el puerto 12345, impleméntelo en Automation Manager e inícielo. WSDL está disponible en:

```
http://localhost:12345
```

El WSDL expone varios métodos que proporcionan datos al desencadenador. Elija el método más apropiado para lo que está tratando de lograr.

- Los métodos cuyos nombres incluyen *WithResponse* permiten enviar respuestas personalizadas, como mensajes de error personalizados, vistas previas de etiquetas, archivos PDF, archivos de impresión (\*.PRN) y otras opciones similares. Los métodos cuyos nombres no incluyen *WithResponse* en su nombre, igual proporcionarán comentarios, pero no puede personalizar la respuesta. Los comentarios contendrán mensajes de error predeterminados.
- Los métodos que tienen *SetVariables* en su nombre, le permiten proporcionar una lista de variables en dos formatos predefinidos. Automation extrae valores automáticamente y los asigna a las variables apropiadas. Esto le permite ahorrar tiempo porque no es necesario que configure ningún filtro para la extracción y la asignación. Para los métodos cuyos nombres no incluyen *SetVariables* en su nombre, debe definir el filtro.

La interfaz del servicio Web define los siguientes métodos:

### Método ExecuteTrigger

La parte principal de la definición es la siguiente:

```
<wsdl:message name="WebSrviTrg_ExecuteTrigger_InputMessage">
  <wsdl:part name="text" type="xsd:string"/>
  <wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="WebSrviTrg_ExecuteTrigger_OutputMessage">
  <wsdl:part name="ExecuteTriggerResult" type="xsd:int"/>
</wsdl:message>
```

```
<wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>
```

La definición incluye dos variables de entrada (debe proporcionar los valores):

- **text:** El filtro en la configuración analiza esta cadena de entrada. Generalmente, la cadena de entrada está estructurada como archivo CSV o XML, lo que hace que sea fácil de analizar. También puede usar cualquier otro formato de archivo textual.
- **wait:** Este es un archivo booleano que especifica dos cosas:
  - Si desea esperar la respuesta del estado del trabajo de impresión o no.
  - Si el Servicio Web debe proporcionar comentarios o no.

En caso de que sea *Verdadero*, use 1. En caso de que sea *Falso*, use 0. Según el tipo de método que seleccione, hay una respuesta predefinida o puede enviar la respuesta personalizada.

Existen las siguientes variables de salida opcionales (puede recibir los valores, si los solicita, configurando **esperar** en 1):

- **ExecuteTriggerResult:** la respuesta del entero contiene el valor 0 en caso de errores informados en el procesamiento de los datos. Contiene un entero mayor a 0 si hay errores. La aplicación que ejecuta la llamada del servicio Web a NiceLabel Automation puede utilizar la respuesta como indicador de error.
- **errorText:** el valor de esta cadena contiene la respuesta del estado del trabajo de impresión si se produce un error durante el procesamiento del desencadenador.



#### NOTA

Si se produjo algún error durante el procesamiento del desencadenador, este elemento se incluye en el mensaje de respuesta XML y su valor contiene la descripción del error. No obstante, si no se produjo ningún error, este elemento no se incluirá en la respuesta XML.

### Método `ExecuteTriggerWithResponse`

Este método se utiliza cuando el desencadenador debe enviar la respuesta personalizada después de que completa la ejecución.

Algunos ejemplos de lo que puede enviar como respuesta personalizada: mensajes de error personalizados, vista previa de etiqueta, archivos PDF generados, archivo de secuencia de impresión (archivo de cola de impresión), archivo XML con detalles del motor de impresión más la vista previa de etiqueta (codificada como cadena Base64); las posibilidades son interminables.

La parte principal de la definición es la siguiente:

```
<wsdl:message name="WebSrviTrg_ExecuteTriggerWithResponse_InputMessage">
  <wsdl:part name="text" type="xsd:string"/>
  <wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="WebSrviTrg_ExecuteTriggerWithResponse_OutputMessage">
```

```

<wsdl:part name="ExecuteTriggerWithResponseResult" type="xsd:int"/>
<wsdl:part name="responseData" type="xsd:base64Binary"/>
<wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>

```

En el ejemplo anterior, hay dos variables de entrada (debe proporcionar los valores):

- **text:** El filtro en la configuración analiza esta cadena de entrada. Generalmente, la cadena de entrada está estructurada como archivo CSV o XML, lo que hace que sea fácil de analizar. También puede usar cualquier otro formato de archivo textual.
- **wait:** Este es un archivo booleano que especifica dos cosas:
  - Si desea esperar la respuesta del estado del trabajo de impresión o no.
  - Si el Servicio Web debe proporcionar comentarios o no.

En caso de que sea *Verdadero*, use 1. En caso de que sea *Falso*, use 0. Según el tipo de método que seleccione, hay una respuesta predefinida o puede enviar la respuesta personalizada.

Además, en el ejemplo anterior se incluyen las siguientes variables de salida opcionales.



#### NOTA

usted recibe valores de variables de salida opcionales si los solicita al configurar el valor del campo **esperar** en 1.

- **ExecuteTriggerWithResponseResult:** la respuesta del entero contiene 0 si no hubo problemas durante el procesamiento de los datos. La respuesta contiene un entero mayor a 0 si hay errores. La aplicación que ejecuta la llamada del servicio Web a NiceLabel Automation puede utilizar la respuesta como indicador de error.
- **responseData:** respuesta personalizada que puede definir en la configuración del desencadenador de servicio Web. La respuesta son datos codificados en base64.
- **errorText:** si surge un error durante el procesamiento del desencadenador, esta cadena contiene el valor de respuesta del estado del trabajo de impresión.



#### NOTA

Si se informa un error durante el procesamiento del desencadenador, el mensaje de respuesta XML incluye el elemento **errorText**. El valor de este elemento contiene la descripción del error. No obstante, si no se produjo ningún error, este elemento no se incluirá en la respuesta XML.

### Método ExecuteTriggerAndSetVariables

La parte principal de la definición es la siguiente:

```

<wsdl:message name="WebSrviTrg_ExecuteTriggerAndSetVariables_InputMessage">
  <wsdl:part name="text" type="xsd:string"/>

```

```

    <wsdl:part name="variableData" type="xsd:string"/>
    <wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>
    <wsdl:message
name="WebSrviTrg_ExecuteTriggerAndSetVariables_OutputMessage">
    <wsdl:part name="ExecuteTriggerAndSetVariablesResult" type="xsd:int"/>
    <wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>

```

En el ejemplo anterior, hay tres variables de entrada (debe proporcionar los valores):

- **text:** El filtro en la configuración analiza esta cadena de entrada. Generalmente, la cadena de entrada está estructurada como archivo CSV o XML, lo que hace que sea fácil de analizar. También puede usar cualquier otro formato de archivo textual.
- **wait:** Este es un archivo booleano que especifica dos cosas:
  - Si desea esperar la respuesta del estado del trabajo de impresión o no.
  - Si el Servicio Web debe proporcionar comentarios o no.

En caso de que sea *Verdadero*, use 1. En caso de que sea *Falso*, use 0. Según el tipo de método que seleccione, hay una respuesta predefinida o puede enviar la respuesta personalizada.

- **variableData:** Es la cadena que contiene los pares *nombre:valor*. El desencadenador leerá todos los pares y asignará los valores provistos a las variables del desencadenador del mismo nombre. Si la variable no existe en el desencadenador, se descarta ese par *nombre:valor*. Cuando proporciona la lista de variables y sus valores en este método, no tiene que definir ninguna extracción de datos con los filtros. El desencadenador realizará todos los análisis.

Hay dos estructuras disponibles para el contenido `variableData`.

### Estructura XML

El desencadenador proporciona variables dentro del elemento raíz `<Variables />` del archivo XML. El nombre de variable incluye el nombre del atributo, mientras que el valor de variable incluye el valor del elemento.

```

<?xml version="1.0" encoding="utf-8"?>
<Variables>
    <variable name="Variable1">Value 1</variable>
    <variable name="Variable2">Value 2</variable>
    <variable name="Variable3">Value 3</variable>
</Variables>

```



## NOTA

Deberá insertar sus datos XML dentro de la sección CDATA. **CDATA**, es decir, **datos de caracteres**, es una sección del contenido del elemento que está marcado para que el analizador interprete los datos XML como datos de caracteres únicamente, no marcas. Como resultado, el desencadenador maneja todo el contenido como datos de caracteres. Por ejemplo, `<element>ABC</element>` se interpreta como `&lt;element&gt;ABC&lt;/element&gt;`. Cada sección CDATA comienza con la secuencia `<![CDATA[` y termina con la secuencia  `]>`. Para resumir, simplemente coloque sus datos XML dentro de esas secuencias.

### Pares nombre-valor

El desencadenador proporciona variables usando una secuencia de texto. Cada par *nombre:valor* viene en su propia línea. El nombre de la variable está a la izquierda del carácter de igual (=), el valor de la variable está a la derecha.

```
Variable1="Value 1"  
Variable2="Value 2"  
Variable3="Value 3"
```

Estas son las variables de salida opcionales:



## NOTA

Usted recibe valores de variables opcionales si los solicita al configurar el valor del campo **esperar** en 1:

- **ExecuteTriggerAndSetVariablesResult:** la respuesta del entero contiene 0 si no hubo problemas durante el procesamiento de los datos. Contiene un entero mayor a 0 si el procesamiento de los datos informa errores. La aplicación que ejecuta la llamada del servicio Web a NiceLabel Automation puede utilizar la respuesta como indicador de error.
- **errorText:** el valor de esta cadena contiene la respuesta del estado del trabajo de impresión si se produce un error durante el procesamiento del desencadenador.



## NOTA

En caso de error en el procesamiento del desencadenador, este elemento se incluye en el mensaje de respuesta XML. Su valor contiene la descripción del error. No obstante, si no se produjo ningún error, este elemento no se incluirá en la respuesta XML.

### Método `ExecuteTriggerAndSetVariablesWithResponse`

Este método se utiliza cuando el desencadenador debe enviar la respuesta personalizada después de que completa la ejecución.

Algunos ejemplos de lo que puede enviar como respuesta personalizada: mensajes de error personalizados, vista previa de etiqueta, archivos PDF generados, archivo de secuencia de impresión (archivo de cola de impresión), archivo XML con detalles del motor de impresión más la vista previa de etiqueta (codificada como cadena Base64); las posibilidades son interminables.

La parte principal de la definición es la siguiente:

```
<wsdl:message name="WebSrviTrg_ExecuteTriggerAndSetVariablesWithResponse_
InputMessage">
  <wsdl:part name="text" type="xsd:string"/>
  <wsdl:part name="variableData" type="xsd:string"/>
  <wsdl:part name="wait" type="xsd:boolean"/>
</wsdl:message>
<wsdl:message name="WebSrviTrg_ExecuteTriggerAndSetVariablesWithResponse_
OutputMessage">
  <wsdl:part name="ExecuteTriggerAndSetVariablesWithResponseResult"
type="xsd:int"/>
  <wsdl:part name="responseData" type="xsd:base64Binary"/>
  <wsdl:part name="errorText" type="xsd:string"/>
</wsdl:message>
```

Hay tres variables de entrada (debe proporcionar los valores):

- **text:** El filtro en la configuración analiza esta cadena de entrada. Generalmente, la cadena de entrada está estructurada como archivo CSV o XML, lo que hace que sea fácil de analizar. También puede usar cualquier otro formato de archivo textual.
- **wait:** Este es un archivo booleano que especifica dos cosas:
  - Si desea esperar la respuesta del estado del trabajo de impresión o no.
  - Si el Servicio Web debe proporcionar comentarios o no.

En caso de que sea *Verdadero*, use 1. En caso de que sea *Falso*, use 0. Según el tipo de método que seleccione, hay una respuesta predefinida o puede enviar la respuesta personalizada.

- **variableData:** Es la cadena que contiene los pares *nombre:valor*. El desencadenador leerá todos los pares y asignará los valores provistos a las variables del desencadenador del mismo nombre. Si la variable no existe en el desencadenador, se descarta ese par *nombre:valor*. Cuando proporciona la lista de variables y sus valores en este método, no tiene que definir ninguna extracción de datos con los filtros. El desencadenador realizará todos los análisis.

Hay dos estructuras disponibles para el contenido variableData.

### Estructura XML

El desencadenador proporciona variables dentro del elemento raíz `<Variables />` del archivo XML. El nombre de variable incluye el nombre del atributo, mientras que el valor de variable incluye el valor del elemento.

```
<?xml version="1.0" encoding="utf-8"?>
<Variables>
```

```
<variable name="Variable1">Value 1</variable>
<variable name="Variable2">Value 2</variable>
<variable name="Variable3">Value 3</variable>
</Variables>
```



## NOTA

Deberá insertar sus datos XML dentro de la sección CDATA. **CDATA**, es decir, **datos de caracteres**, es una sección del contenido del elemento que está marcado para que el analizador interprete los datos XML como datos de caracteres únicamente, no marcas. Como resultado, el desencadenador maneja todo el contenido como datos de caracteres. Por ejemplo, `<element>ABC</element>` se interpreta como `&lt;element&gt;ABC&lt;/element&gt;`. Cada sección CDATA comienza con la secuencia `<![CDATA[` y termina con la secuencia  `]>`. Para resumir, simplemente coloque sus datos XML dentro de esas secuencias.

### Pares nombre-valor

El desencadenador proporciona variables usando una secuencia de texto. Cada par *nombre:valor* viene en su propia línea. El nombre de la variable está a la izquierda del carácter de igual (=), el valor de la variable está a la derecha.

```
Variable1="Value 1"
Variable2="Value 2"
Variable3="Value 3"
```

Estas son las variables de salida opcionales:



## NOTA

Usted recibe valores si los solicita al configurar el valor del campo **esperar en 1**:

- **ExecuteTriggerAndSetVariablesWithResponseResult**: la respuesta del entero contiene 0 si no hubo problemas durante el procesamiento de los datos. Contiene un entero mayor a 0 si el procesamiento de los datos informa errores. La aplicación que ejecuta la llamada del servicio Web a NiceLabel Automation puede utilizar la respuesta como indicador de error.
- **responseData**: respuesta personalizada que puede definir en la configuración del desencadenador de servicio Web. La respuesta son datos codificados en base64.
- **errorText**: el valor de esta cadena contiene la respuesta del estado del trabajo de impresión si se produce un error durante el procesamiento del desencadenador.



## NOTA

En caso de error en el procesamiento del desencadenador, este elemento se incluye en el mensaje de respuesta XML. Su valor contiene la descripción del error. No obstante, si no se produjo ningún error, este elemento no se incluirá en la respuesta XML.

## Otros

Las opciones en la sección **Comentarios del motor de impresión** especifican los parámetros de comunicación que le permiten recibir comentarios del motor de impresión.

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise

- **Impresión supervisada:** Activa el modo de impresión sincronizada. Utilícela siempre que quiera enviar el estado del trabajo de impresión a una aplicación de terceros. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

Las opciones en la sección **Procesamiento de datos** especifican si quiere recortar los datos para que entren en la variable o ignorar las variables de las etiquetas faltantes. En forma predeterminada, notificará los errores e interrumpirá el proceso de impresión si intenta guardar valores muy largos en la variable de la etiqueta o intenta configurar valores para variables de etiquetas no existentes.

- **Ignorar exceso de contenido de variables:** trunca los valores de datos que exceden la longitud de la variable tal lo definido en el diseñador de etiquetas para que se ajusten. Esta opción está en efecto cuando se configuran valores de variables en los filtros, desde los archivos de comandos y cuando se configuran valores de variables de desencadenadores para las variables de etiquetas del mismo nombre.

## Ejemplo

La variable de la etiqueta acepta 5 caracteres como máximo. Con esta opción habilitada, cualquier valor de más de cinco caracteres se trunca a los primeros cinco caracteres. Si el valor es 1234567 ignora los dígitos 6 y 7.

- **Ignorar variables de etiquetas faltantes:** Al imprimir con [archivos de comandos](#) (como un archivo JOB), el proceso de impresión ignora todas las variables que están:
  - especificadas en el archivo de comandos (que usan el comando [SET](#))
  - no definidas en la etiqueta

Lo mismo sucede si define el área de asignación en un filtro para que extraiga todos los pares nombre-valor, pero su etiqueta contiene menos variables.

Al configurar valores de variables de etiquetas no existentes, informo un error. Si esta opción está habilitada la impresión continúa.



Las opciones en la sección **Secuencia de comandos** especifican las posibilidades de las secuencias de comandos.

- **Idioma de secuencias:** Selecciona el idioma de secuencias para el desencadenador. Todas las acciones **Ejecutar secuencia de comandos** que utiliza dentro de un solo desencadenador usan el idioma de secuencia de comandos. seleccionado.

Las opciones en la sección **Guardar datos recibidos** especifican los comandos disponibles para los datos recibidos por el desencadenador.

- **Guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción y guarde los datos recibidos por el desencadenador. La opción **Variable** habilita un nombre de archivo variable. Seleccione una variable que contenga la ruta y el nombre de archivo.
- **Al guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción para guardar los datos en el desencadenador sólo si se produce un error durante la ejecución de la acción. Puede activar esta opción para mantener los datos que causaron el problema listos para resolver el problema más adelante.



#### NOTA

Asegúrese de habilitar el soporte para impresión supervisada. De lo contrario, Automation no puede detectar errores durante la ejecución. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).



#### NOTA

Automation guarda los datos recibidos en un archivo temporal. Este archivo temporal se elimina inmediatamente después de que se completa la ejecución del desencadenador. La variable interna `DataFileName` hace referencia a ese nombre de archivo. Para obtener más información, consulte el tema [Variables internas](#).

## Seguridad

- **Bloquear y cifrar desencadenador:** Habilita la protección del desencadenador. Si lo habilita, el desencadenador queda bloqueado y usted ya no puede editarlo. Esto cifra las acciones. Únicamente el usuario que tenga una contraseña puede desbloquear el desencadenador y modificarlo.

## 3.8. Desencadenador de programador

Para saber más sobre los desencadenadores en general, consulte el tema [Comprender los desencadenadores](#).

<b>EDITION AVAILABILITY:</b>	Automation Easy	Automation Pro	Automation Enterprise

El desencadenador de programador funciona como un temporizador que inicia la ejecución de acciones en su configuración después de un período de tiempo. Use el desencadenador del programador para configurar una ejecución automatizada repetitiva de acciones dependientes de la hora.

El desencadenador del programador es un desencadenador activo. Esto significa que no espera a que el evento cambie, sino que comienza a ejecutar las acciones asignadas no bien finaliza el intervalo de tiempo definido.

### Ejemplo

Uso típico: un sistema ERP produce 6000 archivos de etiquetas de embalaje por día. Automatización imprime las etiquetas y almacena los archivos de etiquetas usados en un directorio exclusivo. Debido a la gran cantidad de archivos de etiquetas usados, la compañía debe configurar una depuración automatizada de los archivos que tienen más de 48 horas. Automatización usa el desencadenador del programador para borrar archivos de etiquetas obsoletas.

El desencadenador del programador contempla automáticamente los cambios en el horario de verano (DST). El desencadenador siempre toma el horario actual del sistema como referencia:

- En la fecha de inicio del DST (verano), el desencadenador se ejecuta en la primera hora después de la hora del horario definido que falta.

### Ejemplo

La hora del desencadenador del programador se configura en las 2.30 h. El reloj salta de las 2.00 h a las 3.00 h. El desencadenador se ejecuta a las 3.00 h en horario de verano.

- El día que cambia el DST (invierno), el desencadenador se ejecuta solo en la primera aparición de la hora definida.

### Ejemplo

La hora del desencadenador del programador se configura en las 2.30 h. El reloj salta de las 3.00 h a las 2.00 h. El desencadenador ya se ha ejecutado. No activa las acciones nuevamente cuando el reloj está en el horario de invierno de las 2.30 h.

## 3.8.1. General

Esta sección le permite configurar los parámetros más importantes del desencadenador.

- **Nombre:** Especifica el nombre único del desencadenador. El nombre lo ayuda a distinguir los diferentes desencadenadores cuando los configura en Automation Builder y luego cuando los ejecuta en Automation Manager.
- **Descripción:** Le permite describir el rol de este desencadenador. Ingrese una breve explicación acerca de lo que hace el desencadenador para ayudar a los usuarios.

## 3.8.2. Recurrencias

Use las recurrencias para definir con qué frecuencia el evento del desencadenador del programador debe repetirse.

- **Ejecutar archivo:** configura el intervalo de tiempo de repetición del desencadenador.
  - **Cada (X) segundos/minutos/horas** configura la hora de repetición del evento del desencadenador en las unidades de tiempo disponibles. Defina la duración del intervalo en **Segundos/Minutos/Horas (X)**.
  - **Diariamente** repite el evento de desencadenador cada día en el horario seleccionado. Defina el horario de repetición diaria en **Hora:**.
  - **En días específicos** repite el desencadenador a la hora definida en el día seleccionado. Configure la repetición usando **Hora:** y días.



### NOTA

Defina los valores de tiempo en formato de 24 horas.

### Otros

Las opciones en la sección **Comentarios del motor de impresión** especifican los parámetros de comunicación que le permiten recibir comentarios del motor de impresión.

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	X	✓	✓

- **Impresión supervisada:** Activa el modo de impresión sincronizada. Utilícela siempre que quiera enviar el estado del trabajo de impresión a una aplicación de terceros. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

Las opciones en la sección **Procesamiento de datos** especifican si quiere recortar los datos para que entren en la variable o ignorar las variables de las etiquetas faltantes. En forma predeterminada, notificará los errores e interrumpirá el proceso de impresión si intenta guardar valores muy largos en la variable de la etiqueta o intenta configurar valores para variables de etiquetas no existentes.

- **Ignorar exceso de contenido de variables:** trunca los valores de datos que exceden la longitud de la variable tal lo definido en el diseñador de etiquetas para que se ajusten. Esta opción está en efecto cuando se configuran valores de variables en los filtros, desde los archivos de comandos y cuando se configuran valores de variables de desencadenadores para las variables de etiquetas del mismo nombre.

### Ejemplo

La variable de la etiqueta acepta 5 caracteres como máximo. Con esta opción habilitada, cualquier valor de más de cinco caracteres se trunca a los primeros cinco caracteres. Si el valor es 1234567 ignora los dígitos 6 y 7.

- **Ignorar variables de etiquetas faltantes:** Al imprimir con [archivos de comandos](#) (como un archivo JOB), el proceso de impresión ignora todas las variables que están:
  - especificadas en el archivo de comandos (que usan el comando [SET](#))
  - no definidas en la etiqueta

Lo mismo sucede si define el área de asignación en un filtro para que extraiga todos los pares nombre-valor, pero su etiqueta contiene menos variables.

Al configurar valores de variables de etiquetas no existentes, informo un error. Si esta opción está habilitada la impresión continúa.

Las opciones en la sección **Secuencia de comandos** especifican las posibilidades de las secuencias de comandos.

- **Idioma de secuencias:** Selecciona el idioma de secuencias para el desencadenador. Todas las acciones **Ejecutar secuencia de comandos** que utiliza dentro de un solo desencadenador usan el idioma de secuencia de comandos. seleccionado.

Las opciones en la sección **Guardar datos recibidos** especifican los comandos disponibles para los datos recibidos por el desencadenador.

- **Guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción y guarde los datos recibidos por el desencadenador. La opción **Variable** habilita un nombre de archivo variable. Seleccione una variable que contenga la ruta y el nombre de archivo.
- **Al guardar datos recibidos por el desencadenador en el archivo:** Habilite esta opción para guardar los datos en el desencadenador sólo si se produce un error durante la ejecución de la acción. Puede activar esta opción para mantener los datos que causaron el problema listos para resolver el problema más adelante.



#### NOTA

Asegúrese de habilitar el soporte para impresión supervisada. De lo contrario, no puede detectar errores durante la ejecución. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).



#### NOTA

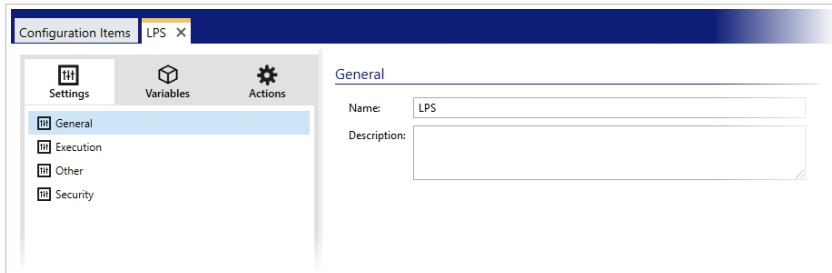
NiceLabel Automation guarda los datos recibidos en un archivo temporal. Este archivo temporal se elimina inmediatamente después de que se completa la ejecución del desencadenador. La variable interna `DataFileName` hace referencia a ese nombre de archivo. Para obtener más información, consulte el tema [Variables internas](#).

## Seguridad

- **Bloquear y cifrar desencadenador:** Habilita la protección del desencadenador. Si lo habilita, el desencadenador queda bloqueado y usted ya no puede editarlo. Esto cifra las acciones. Únicamente el usuario que tenga una contraseña puede desbloquear el desencadenador y modificarlo.

## 3.9. Desencadenador LPS

NiceLabel Automation admite la interfaz de socket TCP/IP del servidor de impresión Loftware (LPS) para entregar los archivos de solicitud de impresión. El protocolo WatchDog (WD) de Loftware maneja la comunicación. Sus aplicaciones entregarán la carga de la misma manera que con LPS, pero Automation procesan datos e imprimen etiquetas.



**El desencadenador LPS** ya está preconfigurado con la acción **Ejecutar archivo de comando** y no necesita configuraciones adicionales. Solo es necesario desplegar el desencadenador.

## 3.10. Probar desencadenadores

Después de configurar los desencadenadores, está en la mitad del proceso de configuración. Antes de implementar el desencadenador, debe probar minuciosamente su funcionamiento previsto cuando ingresan datos y verificar la ejecución de acciones.

Automation Builder Puede probar la configuración mientras continúa trabajando. Algunas acciones tienen capacidades de prueba integradas para que usted pueda concentrarse en la ejecución de la acción individual. También puede probar todos los desencadenadores con el comando Ejecutar vista previa. No obstante, la prueba final siempre debe realizarse en el entorno real, con datos reales y utilizando desencadenadores reales. Así es como monitorea la ejecución del desencadenador, usando Automation Manager.

### Probar ejecución de acciones individuales

Algunas de las acciones incluyen la opción de vista previa. Esto le permite cambiar los parámetros de entrada para ver el resultado de la acción en pantalla.

- **Usar filtro de datos:** La acción ofrece vista previa de los datos analizados. Las reglas del filtro seleccionado se aplican al archivo de datos de entrada seleccionado. El resultado se muestra en la tabla. Si utiliza áreas o subáreas de asignación, puede ver la vista previa para cada nivel de definición del filtro.
- **Ejecutar instrucción SQL:** La acción le permite obtener la vista previa de la ejecución de la instrucción SQL definida. Puede ver el conjunto de datos que resulta de la instrucción SELECCIONAR y la cantidad de filas afectadas por las instrucciones ACTUALIZAR, INSERTAR y ELIMINAR. La ejecución de la vista previa está libre de transacciones, lo que significa puede revertir todos los cambios. Puede cambiar los parámetros de la consulta de entrada y ver cómo influyen en el resultado.
- **Servicio Web:** La acción mostrará la vista previa de la ejecución del método (función) seleccionado del servicio Web. Puede cambiar los parámetros de entrada y ver cómo influyen en el resultado.
- **Ejecutar secuencia de comandos:** La acción comprobará si hay errores de sintaxis en la secuencia de comandos proporcionada y también la ejecutará. Puede cambiar los parámetros de entrada y ver cómo influyen en la ejecución de la secuencia de comandos.

### Probar la ejecución del desencadenador y mostrar la vista previa de la etiqueta en pantalla

Para probar el desencadenador desde el principio, utilice la funcionalidad integrada **Ejecutar vista previa**. Puede ejecutar la vista previa para cada desencadenador, sin importar su tipo. El desencadenador no se disparará al hacer cambios en el evento monitoreado. Solo un desencadenador iniciado en el Automation Manager puede hacerlo. Por el contrario, el desencadenador ejecutará acciones en función de los datos guardados en un archivo. Debe asegurarse de tener un archivo que contenga los datos de muestra que aceptará el desencadenador en la implementación en tiempo real.

El desencadenador ejecutará todas las acciones definidas, incluido el filtrado de datos, y mostrará las vistas previas de etiquetas en pantalla. La vista previa simulará el proceso de impresión en cada detalle. Las etiquetas se imprimirán con la misma composición y el contenido como se muestra en la vista previa. Esto incluye la cantidad de etiquetas y su contenido. Obtendrá información sobre cuántos trabajos de impresión se producen, cuántas etiquetas contiene cada trabajo y la vista previa de cada etiqueta. Puede navegar desde una etiqueta hacia la siguiente en el trabajo de impresión seleccionado.

El panel de registro muestra la misma información como se mostraría en Automation Manager. Expanda las entradas del registro para ver los detalles completos.



#### NOTA

Cuando ejecute la vista previa, se ejecutarán todas las acciones definidas para el desencadenador seleccionado, no solo la acción Ejecutar vista previa. Tenga cuidado cuando utiliza acciones que modificarán datos, como [Ejecutar instrucción SQL](#) o [Servicio Web](#), ya que la ejecución es irreversible.

Para generar vistas previas de etiquetas, realice lo siguiente:

1. Abra la configuración del desencadenador.
2. Asegúrese de que la configuración del desencadenador esté guardada.
3. Haga clic en el botón **Ejecutar vista previa**, ubicado en el grupo Vista previa de la banda de opciones.
4. Examine los archivos de datos que proporcionan el contenido típico que aceptará el desencadenador.
5. Vea el resultado en la pestaña Vista previa.

### **Probar la implementación en el servidor de preproducción**

Es una buena práctica implementar la configuración en Automation Manager en un servidor de preproducción antes de hacerlo en el servidor de producción. Probar en un entorno de preproducción podría identificar problemas de configuración adicionales que no se han detectado durante la prueba del desencadenador en Automation Builder.

También puede probar el esfuerzo de rendimiento agregando la carga al desencadenador para ver cómo responde. La prueba suministrará información importante sobre la capacidad de proceso disponible e identificará los puntos débiles. En función de las conclusiones, puede implementar varias técnicas de optimización del sistema, como modificar el diseño de las etiquetas para producir secuencias de impresión más pequeñas y optimizar el flujo general de datos desde la aplicación existente hacia NiceLabel Automation.






## Diferencias importantes entre la prueba real del desencadenador y la generación de vistas previas en Automation Builder

Si bien las vistas previas del desencadenador en pantalla en Automation Builder ofrecen un método rápido para probar el desencadenador, no debe confiar exclusivamente en ellas. Puede haber diferencias de ejecución entre las vistas previas y la ejecución real del desencadenador cuando utiliza Windows de 64 bits.

Incluso si la configuración funciona correctamente en Automation Builder, asegúrese de ejecutarla en un entorno real utilizando también el Servicio.

- Cuando ejecute el comando **Ejecutar vista previa**, la configuración se ejecutará en Automation Builder, que siempre funciona como aplicación de 32 bits. La generación de vistas previas del desencadenador en Automation Builder solamente prueba la ejecución en una plataforma de 32 bits.
- Cuando ejecute los desencadenadores en un entorno real, la configuración se ejecutará en Servicio, que también funciona como una aplicación de 32 bits en Windows de 32 bits, y se ejecutará como una aplicación de 64 bits en Windows de 64 bits. Para obtener más información, consulte el tema [Ejecutar en modo de servicio](#).
- Pueden presentarse problemas cuando el procesamiento del desencadenador se ve afectado por diferencias de plataforma (32 bits contra 64 bits):
  - **Acceso a base de datos:** Las aplicaciones de 64 bits requieren controladores de bases de datos de 64 bits, y las de 32 bits requieren controladores de 32 bits. Para ejecutar la configuración desde Automation Builder y en el Servicio, necesita controladores de 32 y 64 bits para acceder a su base de datos. Para obtener más información, consulte el tema [Acceso a bases de datos](#).
  - **Sintaxis UNC para archivos de red:** La cuenta de servicio no puede acceder a los archivos de red compartidos con la letra de la unidad asignada. Debe utilizar la sintaxis UNC para archivos de red. Por ejemplo, utilice `\\servidor\recursos compartidos\archivos\etiqueta.lbl` y no `G:\archivos\etiqueta.lbl`, donde G: está asignada a `\\servidor\recursos compartidos`. Para obtener más información, consulte el tema [Acceso a recursos de red compartidos](#).
- Si su servicio NiceLabel Automation se ejecuta con una cuenta de usuario distinta a la que utiliza para Automation Builder, es posible que las cuentas no tengan los mismos privilegios de seguridad. Si puede abrir la etiqueta en Automation Builder, es posible que la cuenta de usuario para el Servicio no tenga acceso. Para ejecutar Automation Builder con la misma cuenta de usuario que la del Servicio, consulte [Usar la misma cuenta de usuario para configurar y ejecutar desencadenadores](#).

## 3.11. Usar capa de transporte seguro (HTTPS)

<b>EDITION AVAILABILITY:</b>	Automation Easy	Automation Pro	Automation Enterprise
			

Puede proteger el tráfico de entrada al [Desencadenador de servidor HTTP](#) y al [Desencadenador de servicio Web](#) activando la compatibilidad con HTTPS. HTTPS asegura la transmisión de los mensajes que se intercambian en la red. La seguridad de la comunicación utiliza certificados X.509 para cifrar los datos que fluyen entre las partes. Su información sigue siendo confidencial a los ojos curiosos porque solo el cliente y NiceLabel Automation pueden descifrar el tráfico. Incluso si algún usuario no autorizado logra interceptar la comunicación, no podrá comprender el significado de los mensajes porque el tráfico aparece como una secuencia de bytes aleatorios.

Una buena práctica de seguridad consisten en cifrar la comunicación en los siguientes casos:

- Usted trabaja con datos sensibles y confidenciales que no deben exponerse a usuarios externos.
- El mensaje debe pasar por redes que están más allá de su control. Por ejemplo, esto sucede cuando envía datos de Automatización por Internet y no desde la red local.

## Habilitar capa de transporte seguro (HTTPS)

Para habilitar el transporte seguro para su desencadenador, haga lo siguiente:

En el sistema Windows:

1. Obtenga el certificado X.509 del emisor de certificados digitales (entidad de certificación - CA). Necesita un tipo de certificado para la "autenticación del servidor".



### NOTA

Si va a autogenerar el certificado, asegúrese de importar el certificado CA en el almacén de Entidades de confianza, para que la firma CA pueda verificarse en el certificado del servidor.

2. Instale el certificado X.509 en el sistema, donde está instalado NiceLabel Automation. Asegúrese de que el certificado sea visible para la cuenta de usuario con la cual ejecuta el servicio NiceLabel Automation. Una buena práctica consiste en instalar el certificado en el almacén del equipo local, no en el almacén del usuario actual. Esto permitirá que NiceLabel Automation utilice el certificado aun cuando no se está ejecutando con la cuenta de usuario con sesión iniciada en ese momento.
  - a. Abra una ventana del Símbolo del sistema.
  - b. Escriba **mmc** y presione la tecla ENTER (asegúrese de estar trabajando con privilegios administrativos).
  - c. En el menú Archivo, haga clic en **Agregar/eliminar complemento**.
  - d. En el cuadro de diálogo **Agregar complemento independiente**, seleccione **Certificados**.
  - e. Haga clic en **Agregar**.
  - f. En el cuadro de diálogo **Complemento de certificados**, seleccione **Cuenta de equipo** y haga clic en **Siguiente**.
  - g. En el cuadro de diálogo **Seleccionar equipo**, haga clic en **Finalizar**.
  - h. En el cuadro de diálogo **Agregar/eliminar complemento**, seleccione **Aceptar**.
  - i. En la ventana Raíz de consola, expanda **Certificados>Personales**.
  - j. Haga clic con el botón derecho en la carpeta Certificados y seleccione **Todas las tareas>Importar**.
  - k. Siga el asistente para importar el certificado.
3. Recupere la huella digital de un certificado que acaba de importar.
  - a. Mientras se encuentra aún en MMC, haga doble clic en el certificado.
  - b. En el cuadro de diálogo **Certificado**, haga clic en la pestaña **Detalles**.
  - c. Desplácese por la lista de campos y haga clic en Huella digital.
  - d. Copie los caracteres hexadecimales del cuadro. Elimine los espacios entre los números hexadecimales. Por ejemplo, la huella digital "a9 09 50 2d d8 2a e4 14 33 e6 f8 38 86 b0 0d 42"

77 a3 2a 7b" debe especificarse como "a909502dd82ae41433e6f83886b00d4277a32a7b" en código. Este **certhash** se requiere en el próximo paso.

4. Vincule el certificado a una dirección IP y puerto donde se está ejecutando el desencadenador. Esta acción habilitará el certificado en el número de puerto seleccionado. Abra el **Símbolo del sistema** (asegúrese de estar trabajando con privilegios administrativos) y ejecute el siguiente comando:

```
netsh http add sslcert
ipport=0.0.0.0:56000 certhash=7866c25377554ca0cb53bcdfd5ee23ce895bdfa2
appid={A6BF8805-1D22-42C2-9D74-3366EA463245}
```

donde:

- **ipport** es el par dirección IP-puerto, donde se ejecuta el desencadenador. Deje la dirección IP como 0.0.0.0 (equipo local), pero cambie el número de puerto para que coincida con el de la configuración del desencadenador.
- **certhash** es la huella digital (hash SHA) del certificado. Este hash tiene 20 bytes de largo y se especifica como una cadena hex.
- **appid** es el GUID de la aplicación propietaria. Puede utilizar cualquier GUID aquí, incluso el del ejemplo de arriba.

En la configuración del desencadenador:

1. En el desencadenador HTTP o de servicio Web, habilite la opción **Asegurar conexión (HTTPS)**.
2. Vuelva a cargar la configuración en Automation Manager.

## Deshabilitar capa de transporte seguro (HTTPS)

En el sistema Windows:

- Desvincule el certificado del par dirección IP-puerto. Ejecute el siguiente comando en el Símbolo del sistema (asegúrese de estar trabajando con privilegios administrativos):

```
netsh http delete sslcert ipport=0.0.0.0:56000
```

donde:

- **ipport** es el par dirección IP-puerto donde se ejecuta el desencadenador y donde se vinculó el certificado.

En la configuración del desencadenador:

1. En el desencadenador HTTP o de servicio Web, deshabilite la opción **Asegurar conexión (HTTPS)**.
2. Vuelva a cargar la configuración en Automation Manager.

## 3.12. Proteger configuración de desencadenadores en edición

La configuración de desencadenadores puede protegerse utilizando dos métodos.

- **Bloqueo del desencadenador.** Mediante este método, bloquea el archivo de configuración del desencadenador y lo protege con una contraseña. Sin la contraseña, nadie puede editar el desencadenador. Habilite la opción **Bloquear y cifrar desencadenador** en *Configuración -> Seguridad* del desencadenador.
- **Configuración de permisos de acceso.** Mediante este método, utiliza los permisos del usuario según se definen en las Opciones de NiceLabel Automation. Puede habilitar grupos de usuario y asignarles distintas funciones. Si al grupo se le asignan privilegios de edición, todos los miembros del grupo pueden editar los desencadenadores. Este método requiere que se habilite el inicio de sesión del usuario. Puede utilizar los usuarios de Windows de grupos locales o directorio activo, o bien definir NiceLabel Automation usuarios. Consulte **Acceso y derechos de usuario** en Configuración.

## 3.13. Configuración del firewall para desencadenadores de red

El desencadenador de red es un desencadenador que se ejecuta usando el protocolo TCP/IP. En Automation, dichos desencadenadores son el desencadenador TCP/IP, el desencadenador HTTP y el

servicio Web. Estos proporcionan servicios de red y están enlazados con la tarjeta de interfaz de red, su dirección IP y el número de puerto configurado. Una vez que implementa e inicia los desencadenadores de red en Automation Manager, comienzan a escuchar al puerto de tráfico de entrada.

Los firewalls protegen a las computadoras de intentos no autorizados de conexiones entrantes. El instalador de NiceLabel se asegura de que las secuencias de comunicación entrantes establecidas con todos los puertos propiedad del servicio de Automation estén permitidas en el Firewall de Windows.



## AVISO

El servicio de Automation posee puertos configurados para los desencadenadores TCP/IP, pero no posee puertos definidos para el desencadenador HTTP y el desencadenador del servicio Web. Estos proyectos están enlazados con el proceso del ID 4 (SYSTEMA) y no con el proceso del Automation Servicio.

Configure el firewall para permitir la comunicación en puertos configurados para desencadenadores HTTP y del servicio Web. Para crear una regla de entrada, siga estos pasos:

1. En la computadora que está ejecutando NiceLabel Automation, en el menú **Inicio**, seleccione **Panel de control**, seleccione **Sistema y seguridad** y seleccione **Firewall de Windows**.
2. En el panel de navegación, seleccione **Configuración avanzada**.
3. En la ventana **Firewall de Windows con seguridad avanzada**, en el panel de navegación seleccione **Reglas de entrada** y luego, en el panel Acciones, seleccione **Nueva regla**.
4. En la página **Tipo de regla**, seleccione **Puerto** y luego haga clic en **Siguiente**.
5. En la página **Protocolo y puertos**, seleccione **Puertos locales específicos** e ingrese el número de puerto en el cual se ejecutan sus desencadenadores HTTP o servicio Web.
6. Haga clic en **Siguiente**.
7. En la página **Acciones**, seleccione **Permitir la conexión** y haga clic en **Siguiente**.
8. En la página **Perfil**, seleccione los perfiles y haga clic en **Siguiente**.
9. En la página **Nombre**, ingrese un nombre para la regla y haga clic en **Finalizar**.

Deben seguirse pasos similares con otros software de firewall.

## 4. Variables

Las variables se usan como contenedores para valores de datos. Necesita las variables para transferir valores a la etiqueta en la acción **Imprimir etiqueta** o usar valores en otras acciones para la manipulación de datos. Generalmente, el filtro extrae valores de las secuencias de datos recibidas por el desencadenador y envía valores a las variables. Para obtener más información, consulte el tema [Comprender los filtros](#).

Generalmente, se recomienda enviar los valores de las variables a la plantilla de la etiqueta y la etiqueta para imprimir. El mecanismo que envía valores de variables a etiquetas usa la asignación de nombre automatizada. El valor de la variable definida en el desencadenador se envía a la variable definida en la etiqueta que lleva el mismo nombre. Puede definir las variables usando uno de los tres métodos disponibles:

- **Importar variables desde archivo de etiquetas:** Para la asignación automática explicada, es una buena práctica importar las variables desde la etiqueta cada vez. Esto le permite ahorrar tiempo y le asegura que los nombres de variables coincidan. La variable importada no solo hereda el nombre de la variable, sino también las propiedades de las variables admitidas, como el valor predeterminado y la longitud.
- **Definir variables manualmente:** Cuando defina manualmente las variables, debe tener mucho cuidado de usar los mismos nombres que las variables de la etiqueta. Deberá definir manualmente las variables que no existan en la etiqueta, pero las necesita dentro del desencadenador.



### NOTA

Un ejemplo pueden ser variables como `LabelName`, `PrinterName`, `Quantity` y variables similares que necesite recordar, el nombre de la etiqueta, el nombre de la impresora, la cantidad u otros meta-valores asignados por el filtro.

- **Habilitar variables internas:** Los valores para las variables internas los asigna NiceLabel Automation y están disponibles como valores de solo lectura. Para obtener más información, consulte el tema [Variables internas](#).



### SUGERENCIA

Si habilita el **área de asignación** (en filtros de texto no estructurado y XML) y la **estructura dinámica** (en filtros de texto estructurado), NiceLabel Automation extraerá los pares **nombre:valor** de los datos del desencadenador y se envían automáticamente los valores a las variables del mismo nombre que están definidas en la etiqueta. No es necesaria ninguna asignación variable manual.

## Propiedades

- **Nombre:** Especifica el nombre de variable único. Los nombres no distinguen mayúsculas de minúsculas. A pesar de que puede usar espacios en los nombres de las variables, no es recomendable. Incluso más si usa variables en cadenas o en condiciones en acciones, porque tendrá que escribirlos entre corchetes.
- **Caracteres permitidos:** Especifica la lista de caracteres que puede ocupar el valor. Puede seleccionar **Todos** (se aceptan todos los caracteres), **Numérico** (solo se aceptan dígitos) o **Binario** (se aceptan todos los caracteres y códigos de control).
- **Limitar longitud de variable:** Especifica el número máximo de caracteres que puede ocupar la variable.
- **Longitud fija:** Especifica que el valor debe ocupar exactamente tantos caracteres como se definen según la longitud.



### NOTA

Es necesario limitar la longitud de variable para ciertos objetos de etiqueta. Un ejemplo es el código de barras EAN-13, que acepta 13 dígitos.

- **Valor requerido:** Especifica que la variable debe tener un valor.
- **Valor predeterminado:** Especifica un valor predeterminado. Si la variable no tiene un valor asignado, se usa siempre el valor predeterminado.

## 4.1. Usar valores compuestos

Algunos objetos en la configuración del desencadenador aceptan valores compuestos. Dicho contenido puede ser una combinación de valores fijos, variables y caracteres especiales (códigos de control). Los objetos que aceptan valores compuestos se identifican con un pequeño botón de flecha a la derecha del objeto. Haga clic en el botón de la flecha para ingresar una variable o un carácter especial.

- **Uso de valores fijos:** Ingrese manualmente un valor fijo para la variable.

```
This is fixed value.
```

- **Uso de valores y datos fijos desde variables:** Puede definir un valor compuesto, combinado de valores variables y fijos. Los nombres de las variables deben estar encerrados entre corchetes [ ]. Puede escribir las variables en forma manual o insertarlas haciendo clic en el botón de la flecha a la derecha. Durante el procesamiento, los valores de las variables se combinarán con los datos fijos y se usarán como contenido. Para obtener más información, consulte el tema [Sugerencias y trucos para usar variables en acciones](#).

En este caso, el contenido combina tres variables y datos fijos ingresados manualmente.

```
[variable1] // This is fixed value [variable2][variable3]
```



- **Uso de caracteres especiales:** Puede agregar caracteres especiales a la combinación. Ingrese los caracteres especiales en forma manual o insertarlos. Para obtener más información, consulte el tema [Introducir caracteres especiales](#).

En este caso, el valor de la `variable1` se combina con datos fijos y caracteres binarios del avance de página.

```
[variable1] Form feed will follow this fixed text <FF>
```

## 4.2. Variables internas

Las variables internas están predefinidas por NiceLabel Automation. Sus valores se asignan automáticamente y están disponibles en el modo de solo lectura. El icono con el símbolo de candado frente al nombre de la variable diferencia las variables internas de las variables definidas por el usuario. Puede usar las variables internas en sus acciones de la misma manera que usaría las variables definidas por el usuario. Las variables internas relacionadas con el desencadenador funcionan como variables internas para cada desencadenador.

Variable interna	Disponible en desencadenador	Descripción
ActionLastErrorDesc	Todos	Brinda la descripción del último error de ocurrió. Use este valor como comentario para el sistema host para identificar la causa del error.
ActionLastErrorID	Todos	Proporciona el ID del último error. Es un valor entero. Cuando el valor es 0, no hubo error. Puede usar este valor en ciertas condiciones, evaluando si hubo un error o no.
BytesOfReceivedData	TCP/IP	Proporciona el número de bytes recibidos por el desencadenador.
ComputerName	Todos	Proporciona el nombre de la computadora en la que se ejecuta la configuración.
ConfigurationFileName	Todos	Proporciona la ruta de acceso y el nombre del archivo de la configuración actual (archivo .MISX).
ConfigurationFilePath	Todos	Proporciona la ruta de acceso del archivo de configuración actual. También consulte la descripción para el ConfigurationFileName.
DataFileName	Todos	Proporciona la ruta de acceso y el nombre del archivo de la copia en funcionamiento de los datos recibidos. Cada vez que el desencadenador acepta los datos, realiza una copia de seguridad de estos en el nombre de archivo único identificado por esta variable.

Base de datos	Base de datos	Proporciona el tipo de base de datos configurada en los desencadenadores.
Fecha	Todos	Proporciona la fecha actual en el formato especificado por la configuración regional del sistema, por ejemplo, 26.2.2018.
DateDay	Todos	Proporciona el número del día actual de un mes, por ejemplo, 26.
DateMonth	Todos	Proporciona el número del mes actual de un año, por ejemplo, 2.
DateYear	Todos	Proporciona el número del año actual, por ejemplo, 2018.
DefaultPrinterName	Todos	Proporciona el nombre del controlador de la impresora, que se define en forma predeterminada.
DriverType	Base de datos	Proporciona el nombre del controlador usado para conectarse a la base de datos seleccionada.
Hostname	TCP/IP	Proporciona el nombre de host del dispositivo/computadora conectado al desencadenador.
HttpMethodName	HTTP	Proporciona el nombre del método que brindó el usuario en la solicitud HTTP, como GET o POST.
HttpPath	HTTP	Proporciona la ruta de acceso definida en el desencadenador HTTP.
HttpQuery	HTTP	Proporciona el contenido de la cadena de consultas según cómo lo recibe el desencadenador HTTP.
NumberOfRowsReturned	Base de datos	Proporciona el número de filas que el desencadenador obtiene de una base de datos.
LocalIP	TCP/IP	Proporciona la dirección IP local a la cual responde el desencadenador. Esto es útil si tiene una máquina de hospedaje múltiple con varias tarjetas de interfaz de red (NIC) y quiere determinar a qué dirección de IP se conectó el cliente. Es útil para escenarios de reemplazo de impresoras.
PathDataFileName	Todos	Proporciona la ruta de acceso en la variable DataFileName, sin el nombre del archivo. También consulte la descripción para el DataFileName.

PathTriggerFileName	Archivo	Proporciona la ruta de acceso en la variable TriggerFileName, sin el nombre del archivo. También consulte la descripción para el TriggerFileName.
Puerto	Servicio TCP/IP, HTTP, Web	Proporciona el número de puerto definido en el desencadenador.
RemoteHttpIp	HTTP	Proporciona el nombre de host del dispositivo/computadora conectado al desencadenador.
Remotelp	Servicio Web	Proporciona el nombre de host del dispositivo/computadora conectado al desencadenador.
ShortConfigurationFileName	Todos	Proporciona el nombre del archivo de configuración, sin una ruta de acceso. También consulte la descripción para el ConfigurationFileName.
ShortDataFileName	Todos	Proporciona el nombre del archivo de la variable DataFileName, sin la ruta de acceso. También consulte la descripción para el DataFileName.
ShortTriggerFileName	Archivo	Proporciona el nombre del archivo de la variable TriggerFileName, sin la ruta de acceso. También consulte la descripción para el TriggerFileName.
SystemUserName	Todos	Proporciona el nombre de Windows del usuario registrado.
TableName	Base de datos	Proporciona el nombre de la tabla como se usa en el desencadenador.
Hora	Todos	Proporciona la hora actual en el formato especificado por la configuración regional del sistema, por ejemplo, 15:18.
Hora	Todos	Proporciona el valor de la hora actual, por ejemplo, 15.
TimeMinute	Todos	Proporciona el valor de los minutos actuales, por ejemplo, 18.
Segundos	Todos	Proporciona el valor de los segundos actuales, por ejemplo, 25.

TriggerFileName	Archivo	Proporciona el nombre del archivo que desencadenó las acciones. Es útil cuando monitorea un conjunto de archivos en la carpeta, para que pueda identificar qué archivo exactamente desencadenó las acciones.
TriggerName	Todos	Proporciona el nombre del desencadenador como lo definió el usuario.
Username	Todos	Proporciona el NiceLabel Automation nombre del usuario registrado actualmente. La variable tiene contenido únicamente si el usuario registrado está habilitado.

### 4.3. Variables globales

Las variables globales son un tipo de variables que se pueden usar en diferentes etiquetas. Las variables globales se definen afuera del archivo de etiqueta y recuerdan el último valor usado.

Las variables globales se definen comúnmente como contadores globales. La variable global proporcionará un valor único para cada etiqueta que necesite un valor nuevo. Se lleva a cabo un bloqueo de archivos que garantiza la unicidad de cada valor.

Las variables globales se definen en el diseñador de etiquetas, solo lo usará NiceLabel Automation. El origen de las variables globales puede configurarse en el diálogo **Opciones (Archivo > Opciones > Variables globales)**.

De manera predeterminada, NiceLabel Automation está configurado para utilizar variables globales desde la computadora local. La ubicación predeterminada es:

```
%PROGRAMDATA%\NiceLabel\Global Variables
```

Las variables globales se definen en los archivos **GLOBAL.TDB** y **GLOBALS.TDB.SCH**.

En entornos de múltiples usuarios, asegúrese de configurar todos los clientes de modo que utilicen el mismo origen de red compartida para las variables globales, o variables globales basadas en el Centro de control.



#### NOTA

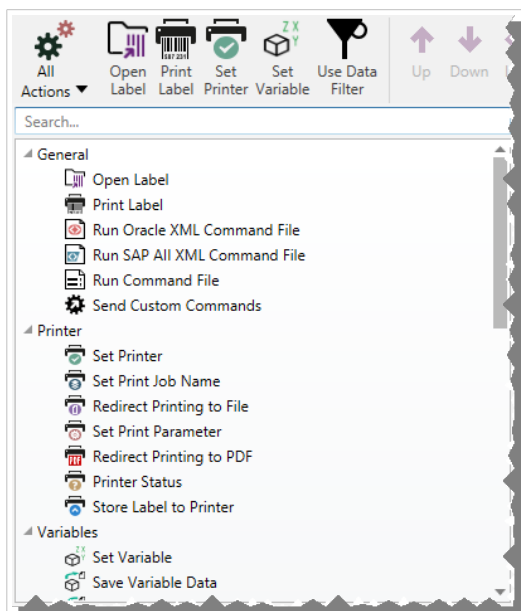
La definición y el valor actual de las variables globales pueden almacenarse en un archivo o en el Centro de control (para los productos **LMS Enterprise** y **LMS Pro**).

# 5. Acciones

La sección Acciones especifica la lista de acciones que se ejecutarán cada vez que se dispare un desencadenador.

## 5.1. Definir acciones

Para definir una acción, haga clic en el icono de acción del grupo de la banda de opciones Insertar acción. La banda de opciones principal contiene las acciones más utilizadas. Para ver todas las acciones disponibles, haga clic en el botón **Todas las acciones**. Para ver los comandos disponibles relacionados con la acción seleccionada, haga clic con el botón derecho en ella y seleccione el comando en la lista.



## 5.2. Acciones anidadas

Son algunas acciones que no pueden utilizarse independientemente. Su funcionalidad específica requiere que estén anidadas debajo de alguna otra acción. Utilice los botones del grupo de la banda de opciones **Orden de las acciones** para cambiar la ubicación de las acciones. Cada acción se identifica con el número de ID que muestra su posición en la lista, incluidas las acciones anidadas. Este número de ID se mostrará en el mensaje de error para que encuentre la acción problemática más fácilmente.



## NOTA

La acción **Imprimir etiqueta** es un buen ejemplo de este tipo de acción. Debe colocarla debajo de la acción **Abrir etiqueta**, de modo que haga referencia a la etiqueta exacta que debe imprimirse.

## 5.3. Ejecución de acciones

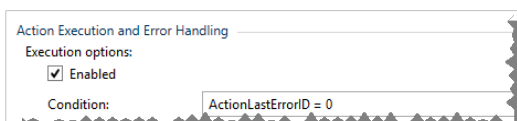
Las acciones de la lista se ejecutarán solo una vez por desencadenador. Como las acciones se ejecutan de arriba abajo, su orden es importante.

Existen dos excepciones. Las acciones **Para bucle** y **Usar filtro de datos** ejecutarán acciones anidadas muchas veces. La acción **Para bucle** se ejecuta tantas veces como esté definido en sus propiedades, y **Usar filtro de datos**, tantas veces como registros haya en un conjunto de datos devueltos por el filtro asociado.

NiceLabel 10 se ejecuta como servicio con una cuenta de usuario de Windows especificada y hereda los permisos de seguridad de la cuenta. Para obtener más detalles, consulte la sección Ejecutar en modo de servicio en la NiceLabel Guía del usuario de Automation.

## 5.4. Acciones condicionales

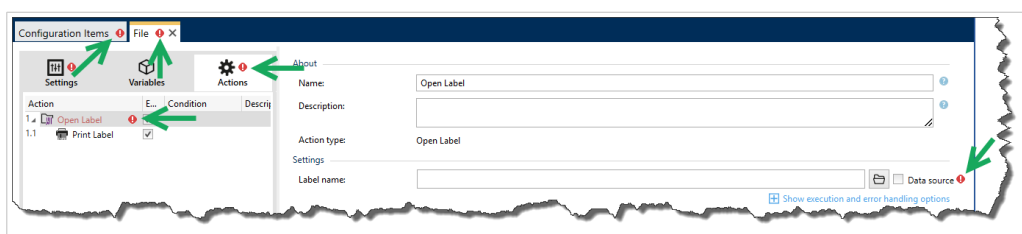
Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones dispuestas permiten su ejecución. La condición es una única secuencia de comandos de línea (VBScript o Python). Para definir la condición, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores** en las propiedades de la acción para expandir las posibilidades.



En este caso, la acción se ejecutará solo si se ha completado correctamente la acción anterior, de modo que la variable interna `ActionLastErrorID` tiene un valor de 0. Para utilizar dicha condición con variables internas, primero debe habilitar la variable interna.

## 5.5. Identificar acciones en el estado de error de configuración

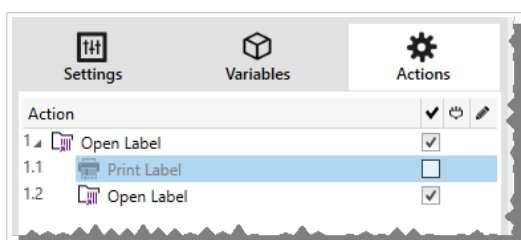
Si una acción no está completamente configurada, se marca con un icono de exclamación rojo. Dichas acciones no pueden ejecutarse. Puede incluirla en la lista **Acción**, pero deberá completar la configuración antes de que pueda iniciar el desencadenador. Si una de las acciones anidadas está en estado de error, todas las flechas de expansión principales (a la izquierda del nombre de la acción) también estarán en rojo para indicar el error de la subacción.



En este caso, la acción **Abrir etiqueta** tiene errores de configuración. No hay ningún parámetro especificado para el nombre de la etiqueta. El icono de exclamación rojo aparece junto al parámetro erróneo en la acción misma, en la lista de acciones, en la pestaña Acciones, en la pestaña del desencadenador y en la pestaña Elementos de configuración. Esto hace que el problema pueda identificarse fácilmente.

## 5.6. Deshabilitar acciones

De manera predeterminada, toda acción recién creada estará habilitada y se ejecutará si se dispara un desencadenador. Puede deshabilitar las acciones que no necesita, pero que igual desea conservar en la configuración. Un acceso directo a la habilitación y deshabilitación de acciones es una casilla de verificación que se encuentra a la derecha del nombre de acción en la lista de acciones definidas.



En este caso, la acción **Establecer impresora** está aún definida en la lista de acciones, pero se ha deshabilitado. Actualmente no es necesaria y se omitirá durante el procesamiento, pero puede volver a habilitarla fácilmente en cualquier momento.

## 5.7. Copiar acciones

Puede copiar la acción y pegarla en el mismo desencadenador o en otro. Puede utilizar los accesos directos convencionales del teclado de Windows o hacer clic con el botón derecho en la acción.

Al hacer clic con el botón derecho en la acción, se mostrarán los comandos contextuales disponibles para el objeto seleccionado.

Automation Builder también le permite hacer una selección de múltiples acciones y realizar las operaciones de copiar, pegar y eliminar con ellas. Para hacer una selección, use Ctrl/Shift + Clic en las acciones requeridas.



### NOTA

Solo es posible seleccionar múltiples acciones en la misma acción principal, es decir todas las acciones seleccionadas deben estar en el mismo nivel.

## 5.8. Navegar en la lista de acciones

Use el mouse para seleccionar la acción definida y luego hacer clic en el botón de flecha respectivo, en el grupo **Orden de acciones** en la banda de opciones. También puede usar el teclado. Las teclas de dirección mueven la selección en la lista de acciones, Ctrl + las teclas de dirección mueven la posición de las acciones hacia arriba y hacia abajo, y también de izquierda a derecha para anidarlas.

## 5.9. Describir las acciones

El grupo **Acerca de** permite describir todas las acciones NiceLabel 10.

- **Nombre:** en forma predeterminada, el nombre de la acción se define según el tipo y, por lo tanto, no es único. Defina un nombre personalizado para poder reconocerla inmediatamente de otras acciones, en los registros y en los mensajes de errores potenciales.
- **Descripción:** notas del usuario para la acción seleccionada. La descripción se muestra en el explorador de acciones.
- **Tipo de acción:** campo de solo lectura que muestra el tipo de acción.



## 5.10. General

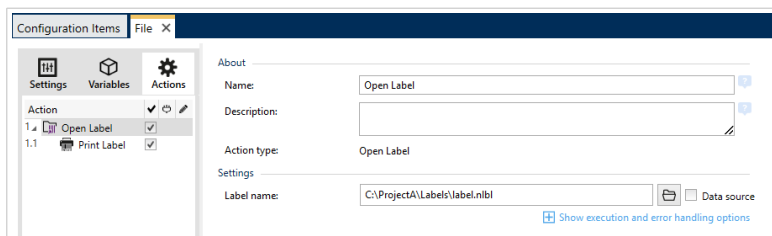
### 5.10.1. Abrir etiqueta

La acción **Abrir etiqueta** especifica el archivo de etiqueta que se imprimirá. Cuando se ejecuta la acción, la plantilla de la etiqueta se abre en la memoria caché. La etiqueta permanece en el caché siempre y cuando la utilicen los desencadenadores o eventos.

No hay límite para la cantidad de etiquetas que pueden abrirse simultáneamente. Si la etiqueta ya está cargada y vuelve a solicitarse, NiceLabel Automation determinará primero si hay una versión más nueva disponible y aprobada para la impresión y luego la abrirá.

En este ejemplo, NiceLabel 10 carga la etiqueta `label.nlbl` desde la carpeta `C:\ProjectA\Labels`.

```
C:\ProjectA\Labels\label.nlbl
```



Si no se encuentra la etiqueta especificada, NiceLabel 10 intenta encontrarlas en ubicaciones alternativas. Para obtener más información, consulte el tema **Combinación de valores en un objeto** en la guía del usuario de NiceLabel Designer.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

#### Usar rutas de acceso relativas

NiceLabel 10 admite el uso de rutas relativas para hacer referencia a su archivo de etiquetas. La carpeta raíz es siempre la carpeta en la que se almacena la solución (o la configuración, en el caso de que la acción se use en una configuración NiceLabel Automation).

Con la siguiente sintaxis, la etiqueta se cargará relativamente desde la ubicación del archivo de configuración. Automation Builder buscará la etiqueta en la carpeta **ProyectoA**, que se encuentra dos niveles arriba de la etiqueta actual, y luego en la carpeta **Etiquetas**.

```
..\..\ProjectA\Labels\label.nlbl
```

El grupo **Configuración** selecciona el archivo de etiqueta.

- **Nombre de etiqueta** especifica el nombre de la etiqueta. Puede integrarse como parte del código, y la misma etiqueta se imprimirá todas las veces. La opción **Origen de datos** permite que el nombre del archivo se defina de forma dinámica. Seleccione o agregue una variable que contenga la ruta de acceso o el nombre del archivo. una vez que se ejecuta un desencadenador o que se lleva a cabo un evento.



### SUGERENCIA

Generalmente, el valor de la variable se asigna según un filtro.



### NOTA

Utilice la sintaxis UNC para los recursos de red.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.10.2. Imprimir etiqueta

Esta acción ejecuta la impresión de etiquetas. Siempre debe anidarse dentro de la acción [Abrir etiqueta](#). El anidamiento permite obtener la referencia a la etiqueta que se imprimirá. Esto a su vez le permite conservar múltiples etiquetas abiertas al mismo tiempo y le permite especificar qué etiqueta se debe imprimir.

Al emitir esta acción, la etiqueta se imprime utilizando el controlador de la impresora definida en la plantilla de la etiqueta. Si el controlador de la impresora definido no se encuentra en el sistema, la etiqueta se imprime utilizando el controlador de la impresora predeterminada del sistema. Puede invalidar el controlador de la impresora utilizando la acción [Establecer impresora](#).

Para lograr una impresión de etiquetas de alto rendimiento, NiceLabel 10 activa dos configuraciones de manera predeterminada:

- **Procesamiento paralelo.** Se realizan múltiples procesos de impresión simultáneamente. La cantidad de subprocesos de impresión en segundo plano depende del hardware, específicamente del tipo de procesador. Cada procesador central puede admitir un único subproceso de impresión. Esta configuración predeterminada puede modificarse. Para obtener más información, consulte la sección Procesamiento paralelo en la guía de usuario NiceLabel Automation.
- **Modo asíncrono.** Tan pronto como se completa el preprocesamiento del desencadenador y las instrucciones para el motor de impresión están disponibles, el subproceso de impresión lo toma en segundo plano. Se devuelve el control al desencadenador para que pueda aceptar la siguiente secuencia de datos de entrada lo antes posible. Cuando el modo sincronizado está habilitado, no se devuelve el control al desencadenador hasta que finaliza el proceso de impresión. Esto puede tardar un tiempo, pero el desencadenador tiene el beneficio de proporcionar comentarios de respuesta a la aplicación que suministra los datos. Para obtener más información, consulte la sección Modo sincronizado en la NiceLabel Automation guía de usuario.



### NOTA

La opción **Guardar error en variable** en **Ejecución de acciones y tratamiento de errores** no producirá ningún resultado en modo asíncrono, ya que el desencadenador no recibirá comentarios desde el proceso de impresión. Para capturar los comentarios del proceso de impresión, primero habilite el modo sincrónico.



## NOTA

Si la acción Imprimir etiqueta está anidada debajo de una acción Para bucle, la Automatización la ejecuta en modo de impresión de sesiones. Este modo funciona como modo de optimización de la impresión que imprime todas las etiquetas en un bucle utilizando un único archivo de trabajo de impresión. Para obtener más detalles, consulte la sección Impresión de sesiones en la guía del usuario NiceLabel Automation.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Cantidad** define el número de etiquetas que se imprimirán utilizando el formulario activo.

- **Etiquetas:** establece el número de etiquetas impresas. **Origen de datos** especifica o agrega una variable de etiqueta que define en forma dinámica la cantidad de etiquetas que se imprimirán.



## NOTA

El valor variable generalmente se asigna mediante la acción **Usar filtro de datos** y debe ser un número entero.

**Todas (cantidad ilimitada):** dependiendo del diseño de la plantilla de etiquetas, las etiquetas se imprimen en distintas cantidades.

### Detalles de cantidad ilimitada de impresión

Generalmente, esta opción se usa en dos escenarios.

1. Se indica a la impresora que imprima continuamente la misma etiqueta hasta que se apague o reciba un comando para borrar su búfer de memoria.



## AVISO

Este escenario requiere que el controlador de impresora Loftware esté instalado y se use para la impresión de etiquetas.

Cuando imprime la etiqueta fija, se envía un solo trabajo de impresión a la impresora, con la cantidad configurada como "ilimitada". Las impresoras de etiquetas tienen un parámetro de comando de impresora que indica impresión "ilimitada".

Cuando la etiqueta no es fija pero incluye objetos que cambian durante la impresión, como los contadores, la cantidad impresa se configurará como la cantidad máxima admitida por la

impresora. El controlador de impresoras de Loftware reconoce el límite de cantidad de la impresora e imprime tantas etiquetas como sea posible.

### Ejemplo

La cantidad máxima de impresiones admitida es 32.000. Esta es la cantidad de etiquetas que se imprimen después de seleccionar la opción Todas (cantidad ilimitada).

2. El desencadenador no proporciona datos, sino que solo actúa como indicador de que "el evento se ha producido". La lógica para adquirir los datos necesarios se encuentra en la etiqueta. Generalmente, se configura una conexión con una base de datos en la etiqueta y en cada desencadenador que la etiqueta debe conectarse con la base de datos, y se adquieren todos los registros de la base de datos. En este caso, la opción **Todas (cantidad ilimitada)** se interpreta como "imprimir todos los registros de la base de datos".
- **Cantidad variable (definida desde la variable de la etiqueta):** especifica una variable de etiqueta que define la cantidad de etiquetas que se imprimirán.  
El desencadenador no recibe el número de etiquetas que se imprimirán, así que pasa la decisión a la plantilla de la etiqueta. La etiqueta puede contener una conexión a una base de datos, que proporcionará la cantidad de etiquetas, o puede haber otro origen para la información de la cantidad. Una variable de la etiqueta debe definirse como "cantidad variable".

El grupo **Avanzadas** define los detalles de impresión de etiquetas. Haga clic en **Mostrar opciones de impresión avanzadas** para definir las opciones de impresión **Avanzadas**:

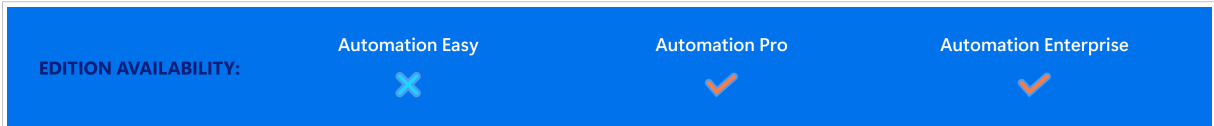
Esta sección especifica ajustes de configuración que no se utilizan frecuentemente relacionados con la cantidad de etiquetas.

- El **Número de etiquetas omitidas** define el número de etiquetas que se omitirán en la primera página de etiquetas. Esta hoja de etiquetas ya puede haberse imprimido una vez, pero no por completo. Puede reutilizar la misma hoja si desplaza la posición de inicio. Puede aplicar esta opción cuando imprime etiquetas en hojas de etiquetas, no en rollos de etiquetas, así que es eficaz para impresoras de oficina, no de etiquetas.
- **Copias de etiquetas idénticas:** especifica el número de copias de etiquetas que se imprimirán para cada registro de la base de datos. Esta opción produce el mismo resultado que la opción principal **Número de etiquetas** cuando tiene etiquetas fijas. Para etiquetas variables, como las que utilizan contadores, puede obtener las copias de etiquetas reales.
- **Conjuntos de etiquetas:** especifica cuántas veces debe repetirse el proceso de impresión de etiquetas.

### Ejemplo

El desencadenador recibirá el contenido con 3 líneas de datos con formato CSV, así que se espera que se impriman 3 etiquetas (1, 2, 3). Si configura esta opción en 3, la impresión se realizará en el siguiente orden: 1, 2, 3, 1, 2, 3, 1, 2, 3.

- **Metadatos:** con cada trabajo de impresión se registran los comentarios de impresión en el Centro de control. Puede ver sus metadatos en **Historial > Impresión > columna de Metadatos de impresión**. Puede utilizar los metadatos para clasificar, filtrar y ejecutar otras funciones en el Centro de control. Los metadatos no afectan la impresión o las secuencias de impresión. Utilice los metadatos para registrar la información adicional sobre sus trabajos de impresión en el Centro de Control. Los metadatos pueden incluir el número LOT u otras variables de etiquetas, los nombres de las impresoras y valores generados por el usuario o el sistema.



### SUGERENCIA

Los valores del grupo Avanzado pueden ser preprogramados o proporcionarse en forma dinámica por una variable existente o recientemente creada.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de

ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

### 5.10.3. Ejecutar archivo de comando XML de Oracle

<b>EDITION AVAILABILITY:</b>	Automation Easy	Automation Pro	Automation Enterprise
	✗	✗	✓

Esta acción ejecuta la impresión con datos de archivos con formato XML de Oracle.

NiceLabel Automation admite internamente archivos XML con la estructura "XML de Oracle", que están definidos por el software Oracle Warehouse Management.

Use esta acción como acceso directo. Ayuda a ejecutar los archivos XML de Oracle sin necesidad de analizarlos con el filtro XML y asignar valores a las variables.

Para poder usar esta acción, el archivo XML debe cumplir con las especificaciones de XML de Oracle. Para obtener más información, consulte la sección Especificaciones XML de Oracle en la guía de usuario NiceLabel Automation.

Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte la sección Acceso a recursos de red compartidos en la guía de usuario NiceLabel Automation.



#### INFORMACIÓN DE NIVEL DE PRODUCTO

Requiere **Automation Builder**.

#### Cómo recibir un archivo de comandos en un desencadenador y ejecutarlo

Cuando el desencadenador recibe el archivo de comandos y quiere ejecutarlo, haga lo siguiente:

1. En el módulo Automation Builder, en la pestaña **Variables**, haga clic en el botón **Variable interna** en la banda de opciones.
2. En la lista desplegable, active la variable interna `DataFileName`. Esta variable interna proporciona la ruta de acceso y el nombre del archivo que contiene los datos recibidos por el desencadenador. En este caso, el contenido es un archivo de comandos. Para obtener más información, consulte la sección Variables internas en la guía de usuario NiceLabel Automation.
3. En la pestaña **Acciones**, agregue la acción para ejecutar el archivo de comandos, tal como Ejecutar archivo de comandos. Para la acción **Ejecutar archivo de comandos**, seleccione **Tipo de archivo**.

4. Active la opción **Variable**.
5. Seleccione la variable **DataFileName** en la lista de variables disponibles.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Archivo** define el comando XML de Oracle que se usará.

- **Nombre de archivo:** archivo de comandos XML de Oracle seleccionado. Pueden ser preprogramados o proporcionarse en forma dinámica usando una variable existente o recientemente creada.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

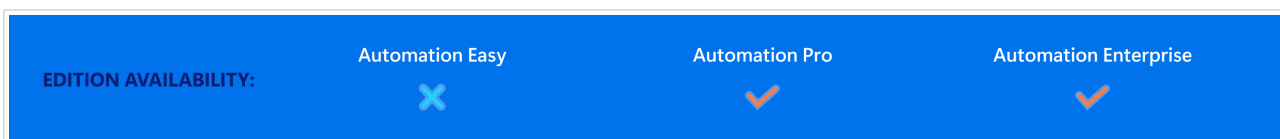
Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de



ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

#### 5.10.4. Ejecutar todo archivo de comando XML de SAP



Esta acción ejecuta la impresión con datos de archivos con formato Todo XML de SAP.

NiceLabel Automation admite internamente archivos XML con la estructura "Todo XML de SAP", que están definidos por el software SAP.

Use esta acción como acceso directo. Esta acción permite ejecutar los archivos Todo XML de SAP directamente sin la necesidad de analizarlos con el filtro XML y asignar valores a las variables. Para poder usar esta acción, el archivo XML debe cumplir con las especificaciones de Todo XML de SAP. Para obtener más información, consulte la sección Especificaciones de Todo XML de SAP en la guía de usuario NiceLabel Automation.

Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte la sección Acceso a recursos de red compartidos en la guía de usuario NiceLabel Automation.



#### INFORMACIÓN DE NIVEL DE PRODUCTO

Requiere **Automation Builder**.

#### Cómo recibir un archivo de comandos en un desencadenador y ejecutarlo

Cuando el desencadenador recibe el archivo de comandos y quiere ejecutarlo, haga lo siguiente:

1. En el módulo Automation Builder, en la pestaña **Variables**, haga clic en el botón **Variable interna** en la banda de opciones.
2. En la lista desplegable, active la variable interna `DataFileName`. Esta variable interna proporciona la ruta de acceso y el nombre del archivo que contiene los datos recibidos por el desencadenador. En este caso, el contenido es un archivo de comandos. Para obtener más información, consulte la sección Variables internas en la guía de usuario NiceLabel Automation.
3. En la pestaña **Acciones**, agregue la acción para ejecutar el archivo de comandos, tal como Ejecutar archivo de comandos. Para la acción **Ejecutar archivo de comandos**, seleccione **Tipo de archivo**.

4. Active la opción **Variable**.
5. Seleccione la variable **DataFileName** en la lista de variables disponibles.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Archivo** define el comando Todo XML de SAP que se usará.

- **Nombre de archivo:** archivo de comandos Todo XML de SAP seleccionado. Pueden ser preprogramados o proporcionarse en forma dinámica usando una variable existente o recientemente creada.

El grupo **Parámetros opcionales** permite definir el nombre de la etiqueta en caso de que no esté incluida en el archivo XML.

- **Nombre de la etiqueta:** el archivo de etiqueta seleccionado que debe usarse en el archivo de comandos. Pueden ser preprogramados o proporcionarse en forma dinámica usando una variable existente o recientemente creada.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.10.5. Ejecutar archivo de comando

Esta acción ejecuta comandos que se incluyen en un archivo de comandos seleccionado. **Todas las opciones Tipo de archivo** proporcionan comandos que NiceLabel 10 se ejecutan en orden de arriba abajo.

Los archivos de comandos generalmente proporcionan datos para una sola etiqueta, pero puede definir archivos de cualquier nivel de complejidad. Para obtener más información, consulte la sección Tipos de archivos de comandos.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Archivo** define el tipo y nombre del archivo de comando que se ejecutará (JOB, XML o CSV).

- **Tipo de archivo.** Especifica el tipo del archivo de comandos que se ejecutará.
- **Nombre de archivo.** Especifica el nombre del archivo de comandos.  
**Nombre de archivo** puede integrarse como parte del código, y el mismo archivo de comandos se ejecutará todas las veces. La opción **Variable** habilita un nombre de archivo variable. Seleccione o cree una variable que contenga la ruta de acceso o el nombre del archivo. una vez que se ejecuta un desencadenador o que se lleva a cabo un evento. Generalmente, el valor de la variable se asigna según un filtro.

Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte la sección Acceso a recursos de red compartidos en la guía de usuario NiceLabel Automation.



## INFORMACIÓN DE NIVEL DE PRODUCTO

Requiere **Automation Builder**.

### Cómo recibir un archivo de comandos en un desencadenador y ejecutarlo

Cuando el desencadenador recibe el archivo de comandos y quiere ejecutarlo, haga lo siguiente:

1. En el módulo Automation Builder, en la pestaña **Variables**, haga clic en el botón **Variable interna** en la banda de opciones.
2. En la lista desplegable, active la variable interna **DataFileName**. Esta variable interna proporciona la ruta de acceso y el nombre del archivo que contiene los datos recibidos por el desencadenador. En este caso, el contenido es un archivo de comandos. Para obtener más información, consulte la sección Variables internas en la guía de usuario NiceLabel Automation.
3. En la pestaña **Acciones**, agregue la acción para ejecutar el archivo de comandos, tal como Ejecutar archivo de comandos.  
Para la acción **Ejecutar archivo de comandos**, seleccione **Tipo de archivo**.
4. Active la opción **Variable**.
5. Seleccione la variable **DataFileName** en la lista de variables disponibles.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.10.6. Enviar comandos personalizados

Esta acción ejecuta los comandos personalizados ingresados de NiceLabel.

Siempre anide esta acción debajo de la acción [Abrir etiqueta](#). Permite hacer referencia a la etiqueta a la cual se aplican los comandos. Para obtener más información, consulte la sección Usar comandos personalizados en la guía de usuario NiceLabel Automation.



## NOTA

La mayoría de los comandos personalizados están disponibles con acciones individuales, de modo que en la mayor parte de los casos no necesita comandos personalizados.



## NOTA

La acción Enviar comandos personalizados puede utilizarse para finalizar el modo Impresión de sesiones. Este modo funciona como modo de optimización de la impresión que imprime todas las etiquetas en un bucle utilizando un único archivo de trabajo de impresión. Para finalizar la impresión de sesiones, anide la acción Enviar comandos personalizados debajo de la acción Para bucle y utilice el comando `SESSIONEND`. Para obtener más detalles, consulte las secciones Impresión de sesiones y Usar comandos personalizados en la guía del usuario NiceLabel Automation.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El editor de **Secuencias de comandos** ofrece las siguientes funciones:

- **Insertar origen de datos:** inserta una variable existente o creada recientemente en la secuencia de comandos.
- **Editor de Secuencias de comandos:** abre el editor, que facilita la creación de secuencias de comandos y la hace más eficiente.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.11. Impresora

### 5.11.1. Establecer impresora

La acción especifica el nombre de la impresora que se utilizará para imprimir la etiqueta activa.



#### NOTA

Esta acción anula la impresora seleccionada en las propiedades de las etiquetas.

Esta acción es útil al imprimir una etiqueta idéntica en varias impresoras. Siempre anide esta acción debajo de la acción [Abrir etiqueta](#) para dar a la etiqueta la referencia para la impresora preferida.

Esta acción lee la configuración predeterminada, como la velocidad y la oscuridad, del controlador de la impresora seleccionada y la aplica a la etiqueta. Si no utiliza la acción **Establecer impresora**, la etiqueta se imprimirá en la impresora según lo definido en la plantilla de la etiqueta.



#### AVISO

Tenga cuidado al cambiar la impresora de una a otra marca, por ejemplo, de Zebra a SATO, o incluso un modelo por otro de la misma marca. La configuración de la impresora puede no ser compatible y es posible que la impresión de la etiqueta no sea idéntica. Asimismo, la optimización del diseño de las etiquetas para la impresora original, como los contadores internos y las fuentes internas, puede no estar disponible en la impresora seleccionada recientemente.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Impresora** especifica el nombre de la impresora que se utilizará para el trabajo de impresión en curso.

- **Nombre de la impresora:** selecciónelo de la lista de controladores de impresoras instalados localmente, o escriba el nombre de la impresora. Seleccione el **Origen de datos** para seleccionar en forma dinámica la impresora usando una variable. Cuando esté activado, seleccione o cree una variable que contenga el nombre de la impresora que se usa cuando se ejecuta la acción.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables **ActionLastErrorId** y **ActionLastErrorDesc**.

### 5.11.2. Establecer nombre de trabajo de impresión

Esta acción especifica el nombre del trabajo de impresión como aparece en el administrador de trabajos de impresión de Windows. Un nombre de trabajo de impresión predeterminado es el nombre del archivo de la etiqueta usado. Esta acción lo anula.





## NOTA

Siempre anide la acción debajo de la acción [Abrir etiqueta](#), de modo que se aplique al archivo de la etiqueta adecuado.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Trabajo de impresión** define el nombre del trabajo de impresión.

- **Nombre:** establece el nombre del trabajo de impresión. Puede tener una codificación rígida, y se utilizará el mismo nombre para cada acción de impresión. Variable habilita un nombre de archivo variable. Seleccione o cree una variable que contenga la ruta de acceso o el nombre del archivo, si se ejecuta un desencadenador o se lleva a cabo un evento.



## NOTA

En Automation Builder, generalmente, el valor de la variable se asigna según un filtro.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

### 5.11.3. Redireccionar impresión a archivo

Esta acción desvía el trabajo de impresión a un archivo. En lugar de enviar el archivo de impresión creado al puerto de una impresora como se define en el controlador de la impresora, la impresión se redirecciona a un archivo. Puede anexar los datos a un archivo existente o sobrescribirlo.

Esta acción le permite capturar los comandos de impresora en un archivo separado.

La acción instruye al módulo de Automation Builder a redireccionar la impresión; como resultado, las etiquetas no se imprimirán. Asegúrese de que la acción esté seguida de la acción [Imprimir etiqueta](#).



## NOTA

NiceLabel se ejecuta como servicio con una cuenta de usuario de Windows. Asegúrese de que esta cuenta de usuario tenga privilegios para acceder a la carpeta especificada con permisos para leer/escribir. Para obtener más información, consulte la sección Acceso a recursos de red compartidos en la guía de usuario de NiceLabel Automation.



## NOTA

La acción Redireccionar la impresión al archivo es útil para imprimir varias etiquetas diferentes (archivos .NLBL) en la impresora de red manteniendo el orden correcto de las etiquetas. Cuando se imprimen múltiples archivos .NLBL del mismo desencadenador, Automation Builder envía cada etiqueta a la impresora en un trabajo de impresión separado, incluso si la impresora de destino es la misma para ambas etiquetas. Cuando se usa una impresora de red, se puede insertar el trabajo de otro usuario entre dos trabajos que el desencadenador deba enviar juntos. Al usar esta acción puede anexas datos de impresión al mismo archivo y luego enviar su contenido a la impresora usando la acción [Enviar datos a impresora](#).

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo de configuración **Archivo** define cómo se realiza la selección del archivo para la redirección.

- **Nombre de archivo:** especifica el nombre del archivo. Pueden ser preprogramados o proporcionarse en forma dinámica usando una variable existente o recientemente creada. Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte la sección [Acceso a recursos de red compartidos](#) en la guía de usuario NiceLabel Automation.



## NOTA

Al utilizar esta acción, asegúrese de que su cuenta de usuario tenga suficientes privilegios para acceder a la carpeta especificada con permisos para leer/escribir.

El grupo de configuración **Modo de escritura de archivo** selecciona el modo en que se trata el archivo en caso de redirecciones repetidas.

- **Sobrescribir el archivo:** si el archivo especificado ya existe en el disco, se sobrescribirá.
- **Anexas datos al archivo:** el archivo del trabajo se agrega a los datos existentes del archivo proporcionado.

El grupo **Persistencia** controla la continuidad de la acción de redirección. Define el número de acciones [Imprimir etiqueta](#) que se ven afectadas por la acción **Redireccionar impresión a archivo**.

- **Aplicar a la próxima acción de impresión:** especifica que la redirección de la impresión se aplique únicamente a la próxima acción [Imprimir etiqueta](#) (evento único).

- **Aplicar a todas las acciones de impresión subsiguientes:** especifica que la redirección de la impresión se aplique a todas las acciones de **Imprimir etiqueta** definidas después de la acción **Redireccionar impresión a archivo** actual.



#### NOTA

La acción únicamente redirecciona la impresión. Asegúrese de que siga la acción [Imprimir etiqueta](#).

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA




Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.11.4. Establecer parámetro de impresión

EDITION AVAILABILITY:	Automation Easy 	Automation Pro 	Automation Enterprise 
-----------------------	--	---	--

Esta acción le permite ajustar los parámetros relacionados con el controlador de la impresora. Entre estos parámetros se incluyen la velocidad y la oscuridad para las impresoras de etiquetas, o la bandeja del papel para las impresoras láser.

La configuración de la impresora se aplica a la impresión en curso únicamente y no se recuerda durante el próximo evento.



### AVISO

Sus parámetros de la acción **Configurar parámetro de impresión** no se aplican cuando se hace una vista previa o una reimpresión de etiquetas desde Control Center.

Para evitar esto, guarde los parámetros de impresión (propiedades de la impresora) en la plantilla de etiqueta o en el controlador de la impresora. Puede que las propiedades de la impresora difieran según el controlador y la impresora.



### NOTA

Si utiliza la acción [Establecer impresora](#) para cambiar el nombre de la impresora, asegúrese de usar la acción **Establecer parámetro de impresión** inmediatamente después. Antes de aplicar la estructura DEVMODE al controlador de la impresora, primero debe cargar la configuración predeterminada del controlador. Esto se realiza con la acción Establecer impresora. DEVMODE únicamente es compatible con DEVMODE del mismo controlador de la impresora.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Establecer parámetros** permite realizar ajustes antes de imprimir.

- **Bandeja de papel:** nombre de la bandeja de papel que contiene los medios de las etiquetas. Esta opción se utiliza comúnmente con impresoras láser y de inyección de tinta con varias bandejas de

papel. El nombre de la bandeja de papel debe coincidir con el nombre de la bandeja del controlador de la impresora. Consulte las propiedades del controlador de la impresora para obtener más detalles.

- **Velocidad de impresión:** define la velocidad de impresión. Esta configuración anula la configuración definida con la etiqueta. El valor provisto debe estar en el rango de valores aceptados.

## Ejemplo

El primer modelo de impresora admite un rango de valores de 0 a 30, mientras que el segundo modelo de impresora admite los valores de -15 a 15. Para obtener más información, consulte las propiedades del controlador de la impresora

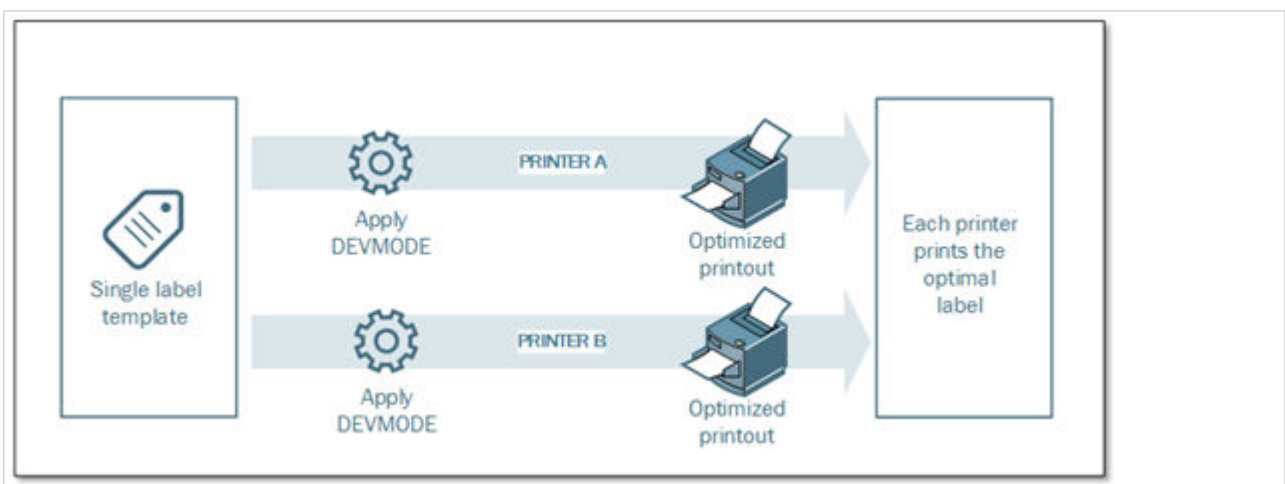
- **Oscuridad:** define la oscuridad de los objetos impresos en el papel y anula la configuración de la etiqueta. El valor provisto debe estar en el rango de los valores aceptados.
- **Desplazamiento de la impresora X:** aplica el desplazamiento horizontal. La impresión de etiquetas se reubicará según el número de puntos específico en la dirección horizontal. Se puede definir un desplazamiento negativo.
- **Desplazamiento de la impresora Y:** aplica el desplazamiento vertical. La impresión de etiquetas se reubicará según el número de puntos específico en la dirección vertical. Se puede definir un desplazamiento negativo.



## SUGERENCIA

Todos los parámetros de impresión pueden ser preprogramados o proporcionarse en forma dinámica usando una variable existente o recientemente creada.

El grupo **Avanzada** personaliza la configuración de la impresora enviada con el trabajo de impresión.



La **configuración de la impresora**, como la velocidad de impresión, la oscuridad, el tipo de medios, el desplazamiento y otros, puede definirse de la siguiente manera:

- Definido en la etiqueta

- Recuperado de un controlador de impresora
- Recuperado de una impresora al momento de la impresión

Los métodos admitidos dependen de las capacidades de la impresora y del controlador de la impresora. El modo de impresión (recuperar configuración de la etiqueta, del controlador o de la impresora) se configura en el diseño de la etiqueta. Es posible que necesite aplicar esta configuración de la impresora al momento de la impresión, pueden variar con cada impresión.

## Ejemplo

Una etiqueta simple debe imprimirse usando una variedad de impresoras, pero cada impresora necesita parámetros ligeramente distintos. Las impresoras de varios fabricantes no utilizan los mismos valores para configurar la velocidad o la temperatura de impresión. Además, algunas impresoras necesitan desplazamiento vertical u horizontal para imprimir la etiqueta en la posición correcta. Durante la etapa de prueba, puede determinar la configuración óptima para cada impresora que quiera utilizar y aplicarla a una plantilla de etiqueta simple justo antes de imprimir. Esta acción aplicará la configuración correspondiente a cada impresora definida.

Esta acción espera recibir la configuración de la impresión en una estructura DEVMODE. Es una estructura de datos estándar de Windows con información sobre la inicialización y el entorno de una impresora.

La opción **Configuración de la impresora** aplica la configuración de la impresora predeterminada. Están disponibles las siguientes entradas:

- **DEVMODE codificada en Base64 con datos fijos.** En este caso, brinde la cadena codificada en Base64 DEVMODE de la impresora directamente en el campo de edición. Al ejecutar esta acción, se convierten los datos codificados en Base64 nuevamente a la forma binaria.
- **DEVMODE codificada en Base64 con datos variables.** En este caso, el origen de datos seleccionado debe tener la estructura DEVMODE codificada en Base64. Habilite Origen de datos y seleccione la variable adecuada de la lista. Al ejecutar esta acción, se convierten los datos codificados en Base64 nuevamente a la forma binaria.
- **DEVMODE binaria con datos variables (disponible en Automation Builder).** En este caso, la variable seleccionada debe tener la estructura DEVMODE en su forma binaria nativa. Habilite **Origen de datos** y seleccione la variable adecuada de la lista. Al ejecutar esta acción, se utilizará la estructura DEVMODE como está, sin ninguna conversión.



### NOTA

Si la variable no brinda la estructura DEVMODE binaria, asegúrese de que la variable seleccionada esté definida como variable binaria en la configuración.



### NOTA

Asegúrese de que la acción [Establecer impresora](#) esté definida delante de esta acción.

La **Configuración de la etiqueta** anula las propiedades de etiquetas definidas en **Propiedades de etiqueta** en Designer. Use esta opción cuando imprima sus etiquetas en una impresora o medio con diferentes propiedades como aparecen en **Propiedades de etiqueta** en Designer. Con esta opción puede:

- Cambiar las dimensiones (altura, ancho) de su etiqueta.
- Agregar o cambiar márgenes en la etiqueta.
- Desactivar el cortador.
- Desactivar la impresión por lotes.
- Aplicar diferentes papeles cambiando el parámetro **Etiqueta horizontal** (conteo horizontal y vertical, espacios, orden de procesamiento).
- Redefinir la orientación como paisaje o retrato.
- Girar las etiquetas 180°.

La automatización aplica la **Configuración de la etiqueta** al momento de imprimir. Los parámetros de la configuración de la etiqueta no se guardan en las plantillas de etiquetas. Puede proporcionar la Configuración de la etiqueta como carga XML.

#### XML de Configuración de etiqueta de muestra

La siguiente muestra presenta una vista estructural de la configuración de la etiqueta y sus atributos.



#### NOTA

Las unidades de medida en el XML coinciden con las unidades de diseño de su etiqueta (cm, in, mm, dot). Puede cambiar las unidades en Designer yendo a **Propiedades de etiqueta > Dimensiones de etiqueta > Unidad de medida**.

```
<LabelSettings>
  <Width>100</Width>
  <Height>30</Height>
  <Margin>
    <Left>2</Left>
    <Right>3</Right>
    <Top>4</Top>
    <Bottom>5</Bottom>
  </Margin>
  <LabelsAcross>
    <Horizontal>
      <Count>2</Count>
      <Gap>4</Gap>
    </Horizontal>
    <Vertical>
      <Count>3</Count>
      <Gap>5</Gap>
    </Vertical>
  </LabelsAcross>
</LabelSettings>
```



```

    </Vertical>
    <ProcessingOrder>HorizontalTopRight</ProcessingOrder>
  </LabelsAcross>
  <Orientation>Landscape</Orientation>
  <Rotated>true</Rotated>
  <DisableCutter/>
  <DisableBatchPrinting/>
</LabelSettings>

```

## Especificación de XML para Configuración de etiqueta

Esta sección contiene una descripción de la estructura del archivo XML para definir los parámetros y valores de la **Configuración de etiqueta**.

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:simpleType name="nonNegativeFloat">
    <xs:restriction base="xs:float">
      <xs:minInclusive value="0"/>
    </xs:restriction>
  </xs:simpleType>
  <xs:element name="LabelSettings">
    <xs:complexType>
      <xs:all>
        <xs:element name="DisableCutter" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:sequence/>
          </xs:complexType>
        </xs:element>
        <xs:element name="DisableBatchPrinting" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:sequence/>
          </xs:complexType>
        </xs:element>
        <xs:element name="Width" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="Height" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
        <xs:element name="Margin" minOccurs="0" maxOccurs="1">
          <xs:complexType>
            <xs:all>
              <xs:element name="Left" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
              <xs:element name="Right" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1"/>
              <xs:element name="Top" type="nonNegativeFloat" minOccurs="0"

```

```

maxOccurs="1" />
    <xs:element name="Bottom" type="nonNegativeFloat" minOccurs="0"
maxOccurs="1" />
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name="LabelsAcross" minOccurs="0" maxOccurs="1">
  <xs:complexType>
    <xs:all>
      <xs:element name="Horizontal" minOccurs="0" maxOccurs="1">
        <xs:complexType>
          <xs:all>
            <xs:element name="Count" type="xs:nonNegativeInteger"
minOccurs="0" maxOccurs="1" />
            <xs:element name="Gap" type="nonNegativeFloat"
minOccurs="0" maxOccurs="1" />
          </xs:all>
        </xs:complexType>
      </xs:element>
      <xs:element name="Vertical" minOccurs="0" maxOccurs="1">
        <xs:complexType>
          <xs:all>
            <xs:element name="Count" type="xs:nonNegativeInteger"
minOccurs="0" maxOccurs="1" />
            <xs:element name="Gap" type="nonNegativeFloat"
minOccurs="0" maxOccurs="1" />
          </xs:all>
        </xs:complexType>
      </xs:element>
      <xs:element name="ProcessingOrder" minOccurs="0" maxOccurs="1">
        <xs:simpleType>
          <xs:restriction base="xs:string">
            <xs:enumeration value="HorizontalTopLeft" />
            <xs:enumeration value="HorizontalTopRight" />
            <xs:enumeration value="HorizontalBottomLeft" />
            <xs:enumeration value="HorizontalBottomRight" />
            <xs:enumeration value="VerticalTopLeft" />
            <xs:enumeration value="VerticalTopRight" />
            <xs:enumeration value="VerticalBottomLeft" />
            <xs:enumeration value="VerticalBottomRight" />
          </xs:restriction>
        </xs:simpleType>
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>

```

```

<xs:element name="Orientation" minOccurs="0" maxOccurs="1">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="Portrait"/>
      <xs:enumeration value="Landscape"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="Rotated" type="xs:boolean" minOccurs="0"
maxOccurs="1"/>
</xs:all>
</xs:complexType>
</xs:element>
</xs:schema>

```

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

### 5.11.5. Redireccionar impresión a PDF

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✘	✘	✔

Esta acción desvía el trabajo de impresión a un documento PDF. El documento PDF creado retiene las dimensiones exactas de la etiqueta, que se definen durante el diseño de la etiqueta. La calidad de representación de los gráficos en el PDF coincide con la resolución de la impresora de destino y el tamaño de impresión deseado.

Los datos de la secuencia de impresión pueden anexarse al archivo existente, o sobrescribir un archivo existente.

La acción instruye a NiceLabel 10 a redireccionar la impresión; como resultado, las etiquetas no se imprimirán. Asegúrese de que la acción esté seguida de la acción **Imprimir etiqueta**.



#### NOTA

El módulo NiceLabel Automation se ejecuta como servicio con una cuenta de usuario de Windows. Asegúrese de que esta cuenta de usuario tenga privilegios para acceder a la carpeta especificada con permisos para leer/escribir. Para obtener más información, consulte la sección Acceso a recursos de red compartidos en la guía de usuario NiceLabel Automation.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Archivo** define el archivo de redirección.

- **Nombre de archivo:** especifica el nombre de archivo hacia el cual desviar el trabajo de impresión. Si está integrado como parte del código, la impresión se redirecciona todas las veces hacia el archivo especificado. Para definirlo en forma dinámica, use una variable existente o cree una nueva.
- **Sobrescribir el archivo:** si el archivo especificado ya existe en el disco, se sobrescribirá (seleccionado de manera predeterminada).

- **Anexar datos al archivo:** el archivo de trabajo se anexa a los datos existentes del archivo en cuestión (no seleccionado de manera predeterminada).
- **Incorporar fuentes en PDF:** Si utiliza alguna fuente no estándar, puede que su solución genere un resultado en PDF diferente en las computadoras que no tengan esa fuente. Si habilita la opción **Incorporar fuentes en PDF**, la fuente no estándar se incorporará en la solución, de modo que la solución genere el mismo PDF en cualquier computadora.

La opción **Incorporar fuentes en PDF** crea un documento conforme a la norma de PDF para archivado (PDF/A). La norma PDF/A garantiza que los documentos se reproduzcan exactamente de la misma manera sin importar software utilizado. La información necesaria para mostrar el contenido, como las imágenes, las fuentes y los datos de colores, se incorporan en el archivo PDF.



#### NOTA

El archivo PDF resultante ocupa más espacio cuando se guarda con esta opción habilitada.

El grupo **Persistencia** permite controlar la persistencia de la acción de redirección. Defina el número de acciones **Imprimir etiqueta** que se ven afectadas por la acción **Redireccionar impresión a archivo**.

- **Aplicar a la próxima acción de impresión:** especifica que la redirección de la impresión se aplique únicamente a la próxima acción **Imprimir etiqueta** (evento único).
- **Aplicar a todas las acciones de impresión subsiguientes:** especifica que la redirección de la impresión se aplique a todas las acciones de **Imprimir etiqueta** definidas después de la acción **Redireccionar impresión a archivo** actual.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

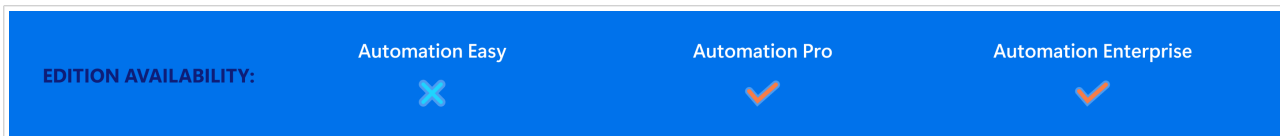
Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.11.6. Estado de impresora



Esta acción se comunica con la impresora para adquirir su estado en tiempo real y se contacta con el administrador de trabajos de impresión de Windows para obtener información adicional sobre la impresora y sus trabajos.

Como resultado, se recopila la información sobre errores, el estado del administrador de trabajos de impresión y la cantidad de trabajos en el administrador. Esto expone potenciales errores y hace que sea fácil identificarlos.



## NOTA

Posibles escenarios de casos de uso. (1) Verificar el estado de la impresora antes de imprimir. Si la impresora está en estado de error, usted imprimirá la etiqueta en una impresora de respaldo. (2) Contar el número de trabajos que se encuentran a la espera en el administrador de trabajos de impresión de la impresora principal. Si hay demasiados, imprimirá la etiqueta en una impresora alternativa. (3) Usted verificará el estado de la impresora antes de imprimir. Si la impresora se encuentra en estado de error, usted no imprimirá la etiquetas, sino que informará sobre el error al sistema principal utilizando alguna de las acciones de salida, como [Enviar datos a puerto TCP/IP](#), [Enviar datos a HTTP](#), [Ejecutar instrucción SQL](#), [Servicio Web](#), o como una respuesta de desencadenador.

## Prerrequisitos del estado en vivo de la impresora

Para que el monitoreo del estado en vivo de la impresora, siga estas instrucciones:

- Utilice el controlador de impresora Loftware para recibir información detallada sobre el estado. Si está usando otro controlador de impresora, solo puede monitorear los parámetros recuperados desde el Administrador de trabajos de impresión de Windows.
- La impresora debe tener capacidad para informar el estado en vivo. Para ver los modelos de impresoras que admiten comunicación bidireccional, consulte la [página web de descargas de NiceLabel](#).
- La impresora debe estar conectada a una interfaz que admita comunicación bidireccional.
- La compatibilidad bidireccional debe habilitarse en **Panel de control > Hardware y sonido > Dispositivos e impresoras > controlador > Propiedades de las impresoras > pestaña Puertos > Habilitar compatibilidad bidireccional**.
- Si utiliza una impresora de etiquetas conectada en red, asegúrese de usar el **Puerto TCP/IP avanzado**, no el **Puerto TCP/IP estándar**. Para obtener más información, consulte [Artículo de base de conocimiento](#).

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Impresora** selecciona la impresora.

- **Nombre de impresora** especifica el nombre de la impresora que se utilizará para el trabajo de impresión en curso. Puede seleccionar una impresora de la lista de controladores de impresoras instalados localmente, o puede escribir el nombre de la impresora. El origen de datos habilita el nombre de la impresora variable. Cuando esté activado, seleccione o cree una variable que contenga el nombre de la impresora cuando se ejecuta un desencadenador o se lleva a cabo un evento. Generalmente, el valor de la variable se asigna según un filtro.

El grupo **Asignación de datos** establece los parámetros que se devuelven como resultado de la acción **Estado de impresora**.



## AVISO

La mayoría de los parámetros siguientes solo son compatibles con controladores de impresora Loftware. Si utiliza cualquier otro controlador de impresora, solamente podrá usar los parámetros relacionados con el administrador de trabajos de impresión.

- **Estado de impresora:** especifica el estado en vivo de la impresora con formato de cadena. Si la impresora informa múltiples estados, todos los estados se combinan en una sola cadena, delimitados por comas ",". Si no se informan problemas de la impresora, este campo estará vacío. El estado de la impresora puede ser **Sin conexión**, **Sin etiquetas** o **Banda de opciones cerca del final**. Como no existe ningún protocolo estándar de informe, cada proveedor de impresoras utiliza sus propios mensajes de estado.
- **Error de impresora:** valor booleano (verdadero/falso) del estado de error de la impresora.
- **Impresora sin conexión:** valor booleano (verdadero/falso) del estado sin conexión de la impresora.
- **Impresora pausada:** valor booleano (verdadero/falso) del estado de pausa de la impresora.
- **Controlador Loftware** valor booleano (verdadero/falso) del estado del controlador de la impresora. Proporciona información si el controlador seleccionado es un controlador Loftware.
- **ID de estado del administrador de trabajos de impresión:** especifica el estado del administrador de trabajos de impresión en forma de cadena, de acuerdo con lo informado por el sistema Windows. El administrador de trabajos de impresión puede informar simultáneamente varios estados. En este caso, los estados se combinan mediante comas ",".
- **ID de estado del administrador de trabajos de impresión:** especifica el estado del administrador de trabajos de impresión en formato de número, de acuerdo con lo informado por el sistema Windows. El administrador de trabajos de impresión puede informar simultáneamente varios estados. En este caso, los ID de estado devueltos contienen todos los ID como indicadores. Por ejemplo, el valor 5 representa los ID de estado 4 y 1, lo cual significa que "La impresora tiene errores, La impresora está pausada". Consulte la siguiente tabla.



### SUGERENCIA

La acción devolverá un valor decimal; los valores de la siguiente tabla se encuentran en hexa, así que tendrá que hacer la conversión antes de analizar la respuesta.

- **Tabla de ID de estados del administrador de trabajos de impresión y descripciones correspondientes**

ID de estado del administrador de trabajos de impresión (en hexa)	Descripción del estado del administrador de trabajos de impresión
0	Sin estado.
1	La impresora está en pausa.
2	La impresora está imprimiendo.
4	Ocurrió un error con la impresora.
8	La impresora no está disponible.
10	La impresora no tiene papel.
20	Se necesita alimentación manual.
40	La impresora tiene un problema con el papel.



80	La impresora está desconectada.
100	Estado de entrada/salida activo.
200	La impresora está ocupada.
400	Atasco de papel.
800	La bandeja de salida está llena.
2000	La impresora está esperando.
4000	La impresora está procesando.
10000	La impresora se está preparando.
20000	El nivel de tóner/tinta es bajo.
40000	No hay tóner en la impresora.
80000	La página actual no se puede imprimir.
100000	Se necesita la intervención del usuario.
200000	La impresora no tiene memoria.
400000	La puerta está abierta.
800000	Error desconocido.
1000000	La impresora está en modo de ahorro de energía.

- **Cantidad de trabajos en el administrador de trabajos de impresión:** especifica la cantidad de trabajos en curso en el administrador de trabajos de impresión para la impresora seleccionada.
- **Estado de impresora extendido del controlador de LoftwareNiceLabel:** devuelve el estado de la impresora y los datos de errores en formato JSON. Los datos incluyen también los parámetros de la impresora, como la versión de firmware, la distancia del cabezal y el nivel de tinta.



#### NOTA

Debe contar con la versión más reciente del controlador de Loftware para obtener el **Estado de impresora extendido del controlador de Loftware**.

### Ejemplo

NiceLabel Automation devuelve el siguiente contenido JSON:

```
{
  "printerName": "Production_printer_09",
  "version": "1",
  "responseType": "status3",
  "overallStatus": "error",
  "operationState": "pause",
  "printerType": "Thermal Transfer Label Printer",
  "deviceStatus":
  [
    {
```

```

        "item": "fatalError",
        "type": "string",
        "internalId": "ERR_PAPER_OUT",
        "shortString": "Paper Out",
        "longString": "Printer is out of paper."
    },
    {
        "item": "error",
        "type": "string",
        "internalId": "ERR_RIBBON_OUT",
        "shortString": "Ribbon Out",
        "longString": "Printer is out of Ribbon."
    }
],
"deviceParameters":
[
    {
        "item": "state",
        "type": "float",
        "internalId": "INFO_PRINTHEAD_DISTANCE",
        "unit": "meter",
        "value": 789,
        "shortDescription": "Printhead distance",
        "longDescription": "Distance of printhead usage."
    },
    {
        "item": "state",
        "type": "string",
        "internalId": "INFO_FW",
        "unit": "",
        "value": "FW27.3.13",
        "shortDescription": "Firmware version",
        "longDescription": "The version of the Firmware loaded on the
printer."
    },
    {
        "item": "property",
        "type": "integer",
        "internalId": "INFO_PRINT_DISTANCE",
        "unit": "seconds",
        "value": 16004,
        "shortDescription": "Print Distance",
        "longDescription": "Print Distance that the printer printed
since odometer reset."
    }
]

```

```
]
}
```

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.11.7. Almacenar etiqueta en impresora

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise

Esta acción guarda una plantilla de etiqueta en la memoria de la impresora. La acción es una parte vital del modo de impresión almacenar/recuperar, con la cual primero almacena una plantilla de etiquetas en la memoria de la impresora y luego la recupera. Las partes que no se cambian del diseño de la etiqueta ya están almacenadas en la impresora, solo tiene que proporcionar los datos para los objetos de etiquetas variables al momento de imprimir. Para obtener más información, consulte la sección Usar modo de impresión almacenar/recuperar en la guía de usuario NiceLabel Automation.



#### NOTA

El tiempo de transferencia de datos de la etiqueta requerida está ampliamente minimizado, ya que hay menos información para enviar. Esta acción se usa generalmente en situaciones de impresión independiente, en las que se almacena la etiqueta en la impresora o el aplicador en la línea de producción y luego se recupera del desencadenador del mismo software o hardware, como el escáner de códigos de barras o la fotocélula.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Opciones avanzadas para almacenar etiqueta en impresora** le permite seleccionar una etiqueta y la variante de almacenamiento preferida.

- **Nombre de la etiqueta que se usará en la impresora:** especifica el nombre que se usará para almacenar la plantilla de la etiqueta en la memoria de la impresora. Ingrese el nombre manualmente o habilite **Origen de datos** para definir el nombre dinámicamente usando un valor variable existente o recientemente creado.



#### AVISO

Al almacenar la etiqueta en una impresora, se recomienda dejar vacío el nombre de la etiqueta debajo de las opciones avanzadas. De esta manera, se evitan conflictos con el nombre de la etiqueta durante el proceso de recuperación de etiquetas.

- **Variante de almacenamiento:** define la ubicación en la memoria de la impresora para las plantillas de etiquetas almacenadas. Ingrese la ubicación manualmente o habilite **Origen de datos** para definir el nombre dinámicamente usando un valor variable existente o recientemente creado.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

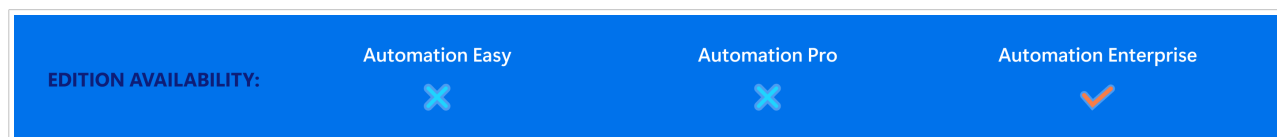
Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.11.8. Imprimir documento PDF



La acción Imprimir documento PDF imprime documentos PDF estáticos que no están vinculados a las etiquetas en sus PowerForms soluciones o NiceLabel Automation configuraciones. Utilice esta acción para imprimir cualquier documento PDF directamente desde sus soluciones o configuraciones. Los documentos PDF se pueden almacenar en:

- Su computadora
- NiceLabel Control Center

- Servidor web
- Unidades de red compartidas



### SUGERENCIA

Esta acción es útil si planea equipar sus paquetes con informes en PDF impresos sobre los elementos incluidos, o si desea imprimir la documentación del paquete sin abrir el explorador de archivos.



### NOTA

Cuando está en uso, la acción Imprimir documento PDF toma un puesto de impresora de la cuota de su licencia. Lea el [NiceLabel documento sobre licencias](#) para obtener más información sobre licencias.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Impresora** especifica el nombre de la impresora que se utilizará para el trabajo de impresión en curso.

- **Nombre de la impresora:** selecciónelo de la lista de controladores de impresoras instalados localmente, o escriba el nombre de la impresora. Seleccione el **Origen de datos** para seleccionar en forma dinámica la impresora usando una variable. Cuando esté activado, seleccione o cree una variable que contenga el nombre de la impresora que se usa cuando se ejecuta la acción.

El grupo **Archivo** define el archivo de redirección.

- **Nombre del archivo:** especifica qué PDF imprimir.



### NOTA

Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte la sección Acceso a recursos de red compartidos en la guía de usuario NiceLabel Automation.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.12. Variables

### 5.12.1. Establecer variable

Esta acción asigna un valor nuevo a la variable seleccionada.

Generalmente, las variables obtendrán los valores con la acción Usar filtro de datos (disponible en Automation Builder), la cual extraerá los campos de los datos recibidos y los asignará a las variables. Además, tal vez necesite configurar usted mismo los valores de las variables, en general, por motivos

de resolución de problemas. En Automation Builder, los valores de las variables no se transmiten entre múltiples desencadenadores, pero sí se mantienen mientras el mismo desencadenador se está procesando.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Variable** define el nombre de variable y su valor.

- **Nombre:** nombre de la variable cuyo valor debería cambiar.
- **Valor:** valor que se establecerá para una variable. Pueden definirse en forma manual o dinámica usando una variable existente o recientemente creada.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo




Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de



ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.12.2. Guardar datos de variable

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
			

Esta acción almacena los valores de una sola o de múltiples variables en un archivo asociado.

En el módulo NiceLabel Automation, esta acción permite el intercambio de datos entre desencadenadores. Para leer datos nuevamente en el desencadenador, use la acción Cargar datos de variables.



### SUGERENCIA

Los valores se guardan utilizando el formato CSV, y la primera línea contiene los nombres de las variables. Si las variables contienen múltiples valores, se codifican nuevos caracteres de línea (CR/LF) como `\n\r`.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Configuración** define el nombre del archivo.

- **Nombre de archivo:** archivo en el que se guardarán los datos de variable. Si el nombre está integrado como parte del código, los valores se guardan cada vez en el mismo archivo. Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte la sección Acceso a recursos de red compartidos en la guía de usuario NiceLabel Automation.

El grupo **Si el archivo existe** ofrece opciones adicionales para guardar los valores.

- **Sobrescribir el archivo:** sobrescribe los datos existentes con los datos de variable nuevos. El contenido anterior se pierde.

- **Anexar datos al archivo:** anexa los valores de variable a los archivos de datos existentes.

El grupo **Estructura del archivo** define los parámetros del archivo de datos variables CSV:

- **Delimitador:** especifica el tipo de delimitador (tabulación, punto y coma, coma o carácter personalizado). El delimitador es un carácter que separa los valores de variables almacenados.
- **Calificador de texto:** especifica el carácter que califica el contenido almacenado como texto.
- **Codificación de archivos:** especifica el tipo de codificación de caracteres que se utilizará en el archivo de datos. **Automática** define la codificación automáticamente. De ser necesario, el tipo de codificación preferida puede seleccionarse en la lista desplegable.



### SUGERENCIA

UTF-8 es una buena selección predeterminada.

- **Agregar nombres de variable en la primera fila:** coloca el nombre de la variable en la primera fila del archivo.

El grupo **Variables** define las variables que se leerán de los archivos de datos. Los valores de las variables existentes se sobrescribirán con valores del archivo.

- **Todas las variables:** datos variables de todas las variables del archivo de datos que se lee.
- **Variables seleccionadas:** los datos variables de las variables enumeradas se leen del archivo de datos.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

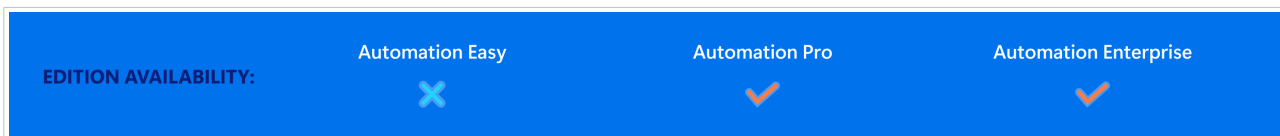
Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.12.3. Cargar datos de variable



Esta acción carga los valores de una o múltiples variables del archivo de datos asociado como se guardaron con la acción **Guardar datos de variable**. Use esta acción para intercambiar datos entre desencadenadores. Puede cargar una variable específica o todas las variables almacenadas en el archivo de datos.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Configuración** define el nombre del archivo.

- **Nombre de archivo:** especifica el archivo desde el que deben cargarse los datos de variable. Si el nombre está integrado como parte del código, los valores se cargan cada vez desde el mismo archivo.

Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte la sección **Acceso a recursos de red compartidos** en la guía de usuario NiceLabel Automation.

La configuración del grupo **Estructura del archivo** debe reflejar la estructura del archivo guardado mediante la acción [Guardar datos de variable](#).

- **Delimitador:** especifica el tipo de delimitador (tabulación, punto y coma, coma o carácter personalizado). El delimitador es un carácter que separa los valores.
- **Calificador de texto:** especifica el carácter que califica el contenido como texto.
- **Codificación de archivos:** especifica el tipo de codificación de caracteres que se utiliza en el archivo de datos. **Automática** define la codificación automáticamente. De ser necesario, seleccione el tipo de codificación preferida en la lista desplegable.



### SUGERENCIA

UTF-8 es una buena selección predeterminada.

El grupo **Variables** define las variables cuyos datos se leerán desde los archivos de datos.

- **Todas las variables:** especifica todas las variables definidas en el archivo de datos que se leerá.
- **Variables seleccionadas:** especifica la selección de variables específicas que se leerán desde el archivo de datos.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.12.4. Manipulación de cadenas

Esta acción define la forma en que deben formatearse los valores de las variables seleccionadas.

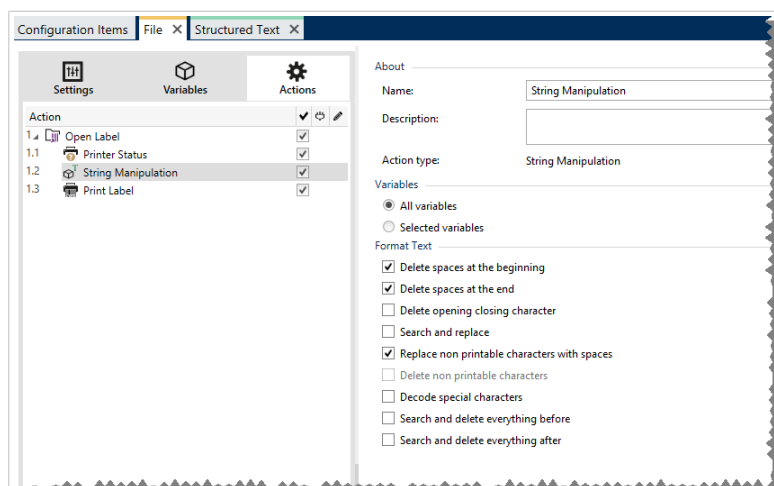
Las acciones de manipulación de cadenas más conocidas son: eliminar espacios iniciales y finales, buscar y reemplazar caracteres y eliminar comillas de apertura y de cierre.

Esta característica a menudo se requiere si un desencadenador recibe un archivo de datos no estructurados o datos heredados. En estos casos, los datos deben analizarse usando el filtro **Datos no estructurados**. La acción de manipulación de cadenas le permite ajustar el valor de los datos.



### NOTA

Si esta acción no proporciona suficiente poder de manipulación de cadenas en un caso en particular, use la acción **Ejecutar secuencia de comandos** para manipular los datos usando Visual Basic Script o Python scripts.



El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Variables** define las variables cuyos valores deben formatearse.

- **Todas las variables:** especifica todas las variables definidas en el archivo de datos al que se le aplicará el formato.
- **Variables seleccionadas:** especifica la selección de variables a las que se aplicará el formato del archivo de datos.

El grupo **Formato de texto** define las funciones de manipulación de cadenas que se aplican a las variables o campos seleccionados. Se pueden utilizar varias funciones. Las funciones se aplican en el mismo orden del editor, de arriba abajo.

- **Borrar espacios en el comienzo:** borra todos los caracteres de espacio (código 32 ASCII decimal) en el comienzo de la cadena.
- **Borrar espacios en el final:** borra todos los caracteres de espacio (valor 32 ASCII decimal) en el final de una cadena.
- **Borrar caracteres de apertura y cierre:** borra los caracteres de apertura y cierre seleccionados que aparecen por primera vez en la cadena.

## Ejemplo

Al utilizar "{" como carácter de apertura y "}" como carácter de cierre, la cadena de entrada `{{selección}}` se convertirá en `{selección}`.

- **Buscar y reemplazar:** ejecuta la función estándar de buscar y reemplazar a partir de los valores provistos para buscar qué y reemplazar con. Se admiten expresiones comunes.



### NOTA

Existen distintas implementaciones de expresiones comunes. NiceLabel 10 utiliza la sintaxis de .NET Framework para las expresiones comunes. Para obtener más información, consulte [Artículo de base de conocimiento](#).

- **Reemplazar caracteres que no se pueden imprimir con espacios:** reemplaza todos los caracteres de control en la cadena con espacios (código 32 ASCII decimal). Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.

- **Borrar caracteres que no se pueden imprimir:** borra todos los caracteres de control de la cadena. Los caracteres que no se pueden imprimir son caracteres con valores ASCII decimales entre 0 y 31, y 127 y 159.
- **Descodificar caracteres especiales:** descodifica los caracteres (o códigos de control) que no están disponibles en el teclado, como Retorno de carro y Avance de línea. NiceLabel 10 utiliza una notación para codificar dichos caracteres en lenguaje natural, como <CR> para Retorno de carro y <LF> para Avance de línea. Esta opción convierte caracteres especiales de la sintaxis NiceLabel en caracteres binarios reales.

### Ejemplo

Cuando reciba los datos "<CR><LF>", Designer utilícelos como una cadena simple de 8 caracteres. Deberá habilitar esta opción para interpretar y usar los datos recibidos como dos caracteres binarios CR (Retorno de carro – ASCII código 13) y LF (Avance de línea – ASCII código 10).

- **Buscar y borrar todo antes de:** busca la cadena provista y borra todos los caracteres que preceden a la cadena definida. La cadena también se puede borrar.
- **Buscar y borrar todo después de:** busca la cadena provista y borra todos los caracteres que están detrás de la cadena definida. La cadena también se puede borrar.
- **Cambiar grafía:** Cambia todos los caracteres de las cadenas a mayúsculas o minúsculas.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.13. Impresión por lotes

### 5.13.1. Para bucle

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✗	✗	✓

Esta acción ejecuta todas las acciones subordinadas (anidadas) múltiples veces. Todas las acciones anidadas se ejecutarán en bucle todas las veces definidas por la diferencia entre el valor de inicio y el valor de finalización.



### NOTA

La acción Para bucle inicia el modo de impresión de sesiones, un modo de optimización de la impresión que imprime todas las etiquetas en un bucle utilizando un único archivo de trabajo de impresión. Para obtener más detalles, consulte la sección Impresión de sesiones en la guía del usuario NiceLabel Automation.



El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Configuración de bucle** incluye las siguientes opciones:

- **Valor de inicio:** referencia de punto de inicio del bucle. Seleccione Origen de datos para definir el valor de inicio en forma dinámica usando un valor de variable. Seleccione o cree una variable que contenga un valor numérico para el inicio.
- **Valor de finalización:** referencia de punto de finalización. Seleccione Origen de datos para definir el valor de inicio en forma dinámica usando un valor de variable. Seleccione o cree una variable que contenga un valor numérico para el inicio.



#### SUGERENCIA

Se permiten valores negativos para el **Valor inicial** y el **Valor final**.

- **Guardar valor de bucle en una variable:** guarda el valor del paso del bucle actual en una variable existente o nueva. Se permite que el valor del paso del bucle contenga algún valor entre el valor de inicio y de finalización. Guarde el valor a fin de reutilizarlo en otra acción para identificar la iteración.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.13.2. Usar filtro de datos

Esta acción ejecuta las reglas de filtros en el origen de datos de entrada. Como resultado, la acción extrae campos de los datos de entrada y asigna sus valores a las variables vinculadas.

La acción Usar filtro de datos ejecuta el filtro seleccionado y asigna a las variables los valores respectivos.

- **Elementos en el nivel inferior:** La acción puede crear elementos en subniveles, identificados con "para cada línea" o "para cada bloque de datos en...". Al verlos, el filtro extraerá los datos, pero no a nivel del documento (con posición de campo que no se puede modificar), sino relativamente de las subáreas que contienen secciones repetidas. En este caso, asegúrese de colocar las acciones debajo de dichos elementos. Debe anidar la acción debajo de esos elementos.
- **Asignación de variables a campos:** La asignación entre variables de desencadenadores y campos de filtro se define manual o automáticamente, según cómo se configure el filtro. Si cuenta con campos definidos manualmente en el filtro, también debe asignar en forma manual los campos a la variable correspondiente.



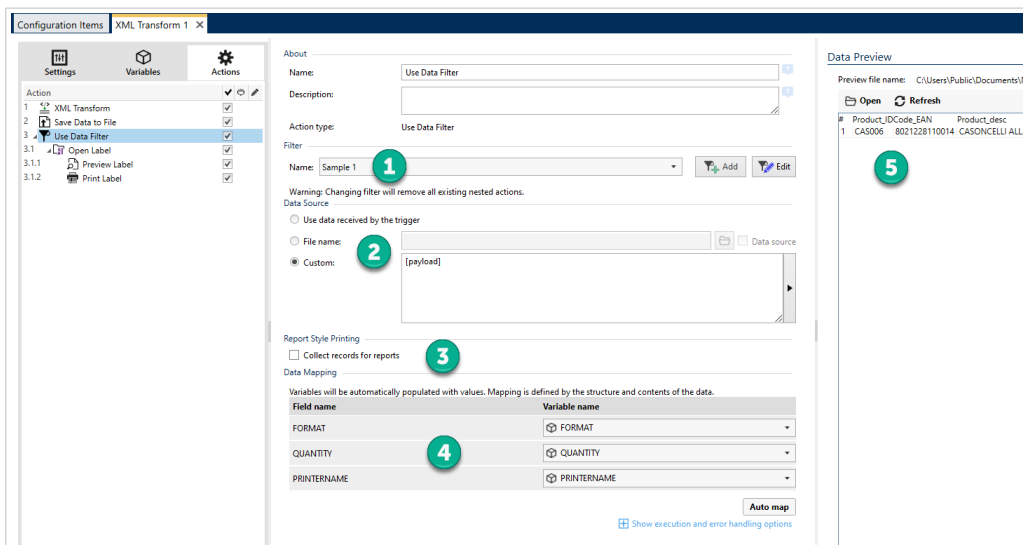
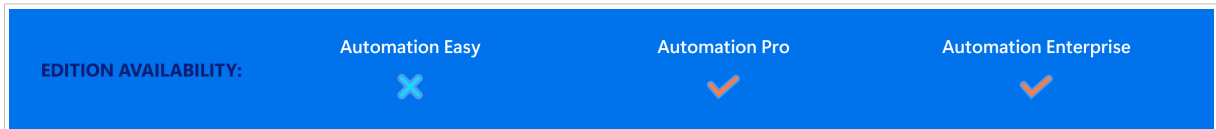
## NOTA

Es una buena práctica definir los campos usando los mismos nombres que los de las variables de etiquetas. En este caso, el botón **Asignar automáticamente** asigna los nombres correspondientes en forma automática.

- **Probar la ejecución del filtro:** Una vez asignadas las variables a los campos, puede probar la ejecución del filtro. El resultado se mostrará en pantalla en la tabla. El número de líneas de la tabla representa el número de veces que se ejecutará la acción en el nivel seleccionado. Los nombres de las columnas representan los nombres de las variables. Las celdas tienen los valores tal cual se

asignaron a la variable respectiva con el filtro. El nombre de archivo predeterminado de la vista previa se hereda de la definición del filtro; también se puede ejecutar el filtro en otro archivo.

- La opción **Recopilar registros para informes** recopila los datos para poder utilizar filtros en la creación de informes. Para obtener más información, consulte el tema [Sección 9.7, “Automatización de informes”](#).



Visualización de la interfaz de filtros de datos.

1. Selección de filtro.
2. Los datos que utiliza el filtro para ejecutar reglas.
3. Recopilación de informes de registros.
4. Asignación de campos (del filtro) a las variables (de la etiqueta/desencadenador).
5. Vista previa de ejecución de filtro.

Para obtener más información, consulte la sección Comprender los filtros y la sección Ejemplos en la guía del usuario de NiceLabel Automation.

El grupo **Filtro** le permite seleccionar qué filtro usar.

- **Nombre:** especifica el nombre del filtro que quiere aplicar. Pueden ser preprogramados o proporcionarse en forma dinámica usando una variable existente o recientemente creada. La lista tiene todos los filtros definidos en la configuración actual. Puede usar los tres elementos de la parte inferior de la lista para crear un filtro nuevo.



## NOTA

Al seleccionar otro filtro se eliminan todas las acciones anidadas en esta acción. Si quiere mantener las acciones definidas actualmente, muévalas afuera de la acción **Usar filtro de datos**. Si se pierde accidentalmente alguna acción, **Deshaga** su acción y vuelva a la configuración anterior.

El grupo **Origen de datos** le permite definir el contenido que desea enviar a la impresora.

- **Usar datos recibidos por el desencadenador:** selecciona los datos recibidos por el desencadenador para usar en un filtro. En este caso, la acción usa los datos originales recibidos por el desencadenador y ejecutará las reglas del filtro en él.

## Ejemplo

Si usa un desencadenador de filtro, los datos representan el contenido del archivo monitoreado. Si utiliza un desencadenador de base de datos, los datos son un conjunto de datos tomados de la base de datos. Si usa un desencadenador TCP/IP, los datos son contenido sin procesar recibidos en un socket.

- **Nombre de archivo:** define la ruta y el nombre del archivo que contiene los datos según el cual se ejecutarán las reglas del filtro. El contenido del archivo especificado se usa en un filtro. La opción **Origen de datos** habilita el nombre de archivo variable. Debe seleccionar o crear una variable que contenga la ruta de acceso o el nombre del archivo.
- **Personalizado:** define el contenido personalizado que se analizará con el filtro. Puede utilizar contenido fijo, una combinación de contenido fijo y variable, o contenido únicamente variable. Para insertar contenido variable, haga clic en el botón con flecha que se encuentra a la derecha del área de datos e inserte una variable de la lista. Para obtener más información, consulte la sección Usar valores compuestos en la guía de usuario NiceLabel Automation.

El campo **Vista previa de datos** proporciona una descripción general del proceso de ejecución del filtro una vez que el contenido del nombre de archivo con vista previa se ha leído y se ha aplicado el filtro seleccionado.

Las reglas del filtro extraen campos. La tabla muestra el resultado de la extracción. Cada línea de la tabla representa datos para una única etiqueta. Cada columna representa una variable.

Para poder observar el resultado, configure la asignación de campos con las variables correspondientes. Según la definición del filtro, puede asignar las variables a los campos manual o automáticamente.

- **Vista previa del nombre del archivo:** especifica el archivo que contiene los datos de muestra que se analizarán a través del filtro. Se copia el archivo en vista previa de la definición del filtro. Si cambia el nombre del archivo en vista previa, se guarda el nuevo nombre.
- **Abrir:** selecciona otro archivo sobre el cual ejecutar las reglas de filtro.

- **Actualizar:** vuelve a ejecutar las reglas de filtro sobre el contenido del nombre del archivo en vista previa. El campo **Vista previa de los datos** se actualizará con el resultado.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

### 5.13.3. Para cada registro

Esta acción ejecuta las acciones anidadas subordinadas múltiples veces. Todas las acciones anidadas se ejecutan en un bucle tantas veces como registros haya en la tabla del formulario con una base de datos conectada.

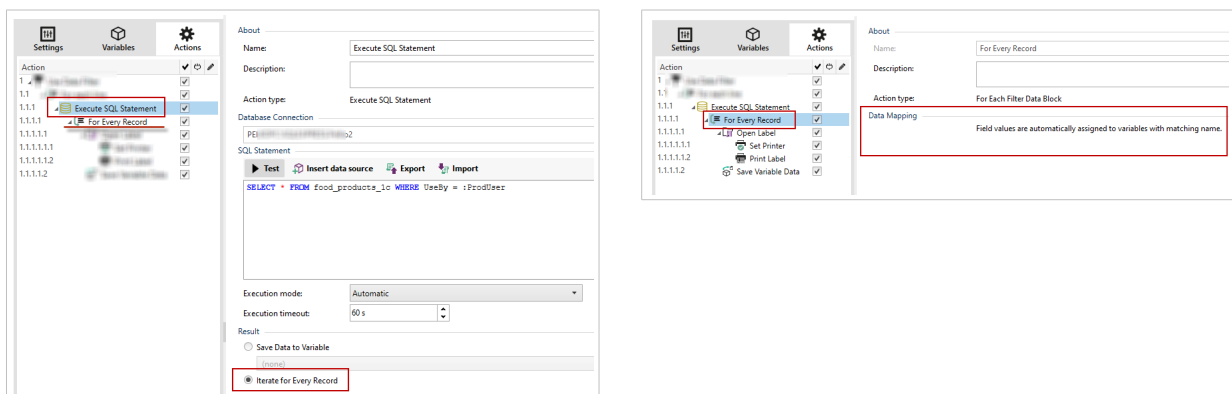
El grupo Acerca de identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Configuración** selecciona los registros.

- **Tabla de formulario:** tabla del formulario que contiene registros para los que debe repetirse una acción.
- **Utilizar todos los registros:** repite una acción para todos los registros de una tabla definida.
- **Utilizar registro seleccionado:** repite una acción únicamente para los registros seleccionados.

Si usa la acción **Ejecutar instrucción SQL** con la opción **Iterar Para cada registro** habilitada, NiceLabel inserta automáticamente la acción **Para cada registro**. Aparecerá una nota sobre la asignación automática de sus variables.



Las variables de solicitud de la etiqueta se conectan automáticamente a los campos de la base de datos que tengan el mismo nombre. Consulte las instrucciones sobre cómo crear sus soluciones:



## IMPORTANTE

Cuando cree su solución con una conexión de base de datos, use las variables de solicitud de sus etiquetas en lugar de los campos de la base de datos.

Para las variables de solicitud, use los mismos nombres que se definieron para los campos de la base de datos. Por ejemplo:

Campo de la base de datos: `food_products_lc.ProdCode`

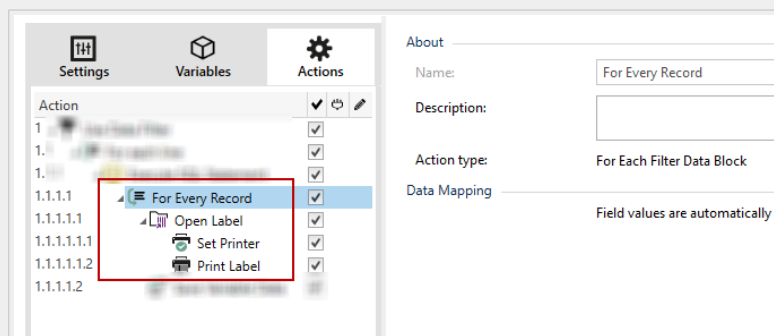
Variable de solicitud de su etiqueta: `ProdCode`

NiceLabel identifica automáticamente las variables que corresponden a los campos de la base de datos.

## Ejemplo

La solución se conecta a la base de datos. Los filtros de configuración solicitaron registros de base de datos y usted desea imprimir los registros filtrados en sus etiquetas.

Use la acción **Para cada registro** y las acciones anidadas para imprimir las etiquetas.



## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de

ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.14. Datos y conectividad

### 5.14.1. Abrir documento/programa

Esta acción proporciona una interfaz con una aplicación externa y se abre usando una línea de comandos.

Las aplicaciones externas pueden ejecutar un procesamiento adicional y devolver los resultados a NiceLabel 10. Esta acción permite que NiceLabel Automation se enlace con cualquier software de terceros que pueda ejecutar procesamientos de datos adicionales o adquirir datos. El software externo puede proporcionar una respuesta de datos y guardarla en un archivo, desde donde puede leerse en variables.



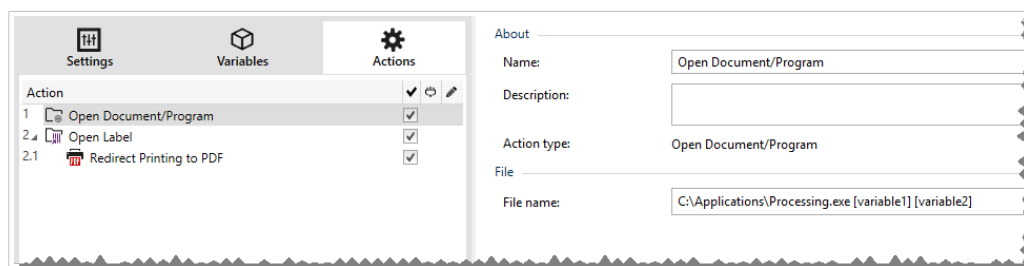
#### NOTA

NiceLabel Automation se ejecuta como una aplicación de servicio y no puede interactuar con el escritorio, aunque se ejecute con los privilegios del usuario que tenga la sesión iniciada. Microsoft impide dicha interacción por motivos de seguridad.

Esto quiere decir que la acción **Abrir documento/programa** de NiceLabel Automation solo puede abrir programas que se ejecuten en segundo plano.

Puede enviar los valores de las variables al programa incluyéndolos en la lista de la línea de comandos entre corchetes.

```
C:\Applications\Processing.exe [variable1] [variable2]
```







## NOTA

Si utiliza esta acción en NiceLabel 10 las soluciones, le permite abrir páginas web o crear mensajes de correo electrónico directamente desde sus formularios. Consulte la sección Creación de hipervínculos y envío de correos electrónicos en formulario en NiceLabel 10 la guía del usuario.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Archivo** define el archivo que se abrirá.

- **Nombre de archivo:** ubicación y nombre de archivo del archivo o aplicación a abrir.  
El nombre de archivo seleccionado puede estar preprogramado, y el mismo archivo se usará cada vez. Si solo se define un nombre de archivo sin ruta, se usa la carpeta con el archivo de configuración (.MISX) de NiceLabel Automation. Puede usar una referencia relativa para el nombre del archivo si se usa la carpeta con el archivo .MISX como carpeta raíz.  
**Origen de datos:** habilita el nombre de archivo variable. Seleccione una variable que contenga la ruta de acceso o el nombre del archivo, o combine varias variables para crear el nombre de archivo. Para obtener más información, consulte la sección Usar valores compuestos en la guía de usuario NiceLabel Automation.



## NOTA

Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte la sección Acceso a recursos de red compartidos en la guía de usuario NiceLabel Automation.

El grupo **Opciones de ejecución** establece los detalles de abertura del programa.

- **Esperar la finalización:** especifica que la ejecución de la acción espere a que se complete esta acción antes de continuar con la siguiente acción programada.



## SUGERENCIA

Habilite esta opción si la acción que sigue depende del resultado de la aplicación externa.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.14.2. Guardar datos en el archivo

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise

Esta acción guarda los valores variables u otras secuencias de datos (como datos binarios) en el archivo. El servicio NiceLabel Automation debe tener acceso de escritura a la carpeta definida.

El grupo **Archivo** define el archivo que se abrirá.

- **Nombre de archivo:** ubicación del archivo que se abrirá en esta acción.

El nombre de ruta y archivo puede estar preprogramado, y el mismo archivo se usará cada vez.

Si solo se define un nombre de archivo sin ruta, se usa la carpeta con el archivo de configuración (.MISX) de NiceLabel Automation. Puede usar una referencia relativa para el nombre del archivo si se usa la carpeta con el archivo .MISX como carpeta raíz.

**Origen de datos:** habilita el nombre de archivo variable. Seleccione una variable que contenga la ruta de acceso o el nombre del archivo, o combine varias variables para crear el nombre de archivo. Para obtener más información, consulte la sección Usar valores compuestos en la guía de usuario NiceLabel Automation.

El grupo **Si el archivo existe** maneja las opciones en caso de un archivo existente.

- **Sobrescribir el archivo:** sobrescribe los datos existentes con los datos nuevos. El contenido anterior se pierde.
- **Anexar datos al archivo:** anexa los valores de variable a los archivos de datos existentes.

El grupo **Contenido** define qué datos se escribirán en el archivo especificado.

- **Usar datos recibidos por el desencadenador:** los datos originales como los recibe el desencadenador se guardarán en el archivo. Efectivamente, esta opción hará una copia de los datos entrantes.
- **Personalizado:** guarda contenido según lo que se indica en el área de texto. Se permiten valores fijos, valores variables y caracteres especiales. Para ingresar variables y caracteres especiales, haga clic en el botón de flecha que se encuentra a la derecha del área de texto. Para obtener más información, consulte la sección Combinación de valores en un objeto en la guía del usuario de NiceLabel Automation.
- **Codificación:** tipo de codificación para los datos enviados. **Automática** define la codificación automáticamente. De ser necesario, seleccione el tipo de codificación preferida en la lista desplegable.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

### 5.14.3. Leer datos desde el archivo

EDITION AVAILABILITY:	Automación Easy ✗	Automación Pro ✓	Automación Enterprise ✓
-----------------------	----------------------	---------------------	----------------------------

Esta acción lee el contenido del nombre del archivo provisto y lo guarda en la variable. Contenido de cualquier tipo de archivo, incluidos los datos binarios que pueden leerse.

Generalmente, el módulo Automation Builder recibe datos para la impresión de etiquetas con el desencadenador. Por ejemplo, cuando se usa el desencadenador del archivo, el contenido del archivo del desencadenador se lee automáticamente y puede ser analizado por filtros. No obstante, puede omitir los filtros para obtener algunos datos externos. Una vez que ejecuta esta acción y tiene los datos almacenados en una variable, puede usar cualquier acción de la variable para usar los datos.

Esta acción es útil:

- Cuando deba combinar datos recibidos por el desencadenador con datos almacenados en algún archivo.



#### AVISO

Si carga datos de archivos binarios (como un archivo de impresión o imagen de mapas de bits), asegúrese de que la variable que almacena el contenido de lectura esté definida como **variable binaria**.

- Cuando quiera intercambiar datos entre desencadenadores. Una vez que el desencadenador prepara los datos y los guarda en un archivo (usando la acción [Guardar datos en el archivo](#)), el otro desencadenador lee los datos.

El grupo **Archivo**: define el archivo desde el que se leerán los datos.

- **Nombre de archivo**: ubicación del archivo que se leerá en esta acción.

El nombre de ruta y archivo puede estar preprogramado, y el mismo archivo se usará cada vez.

Si solo se define un nombre de archivo sin ruta, se usa la carpeta con el archivo de configuración (.MISX) de NiceLabel Automation. Puede usar una referencia relativa para el nombre del archivo si se usa la carpeta con el archivo .MISX como carpeta raíz.

**Origen de datos**: habilita el nombre de archivo variable. Seleccione una variable que contenga la ruta de acceso o el nombre del archivo, o combine varias variables para crear el nombre de archivo. Para obtener más información, consulte la sección Usar valores compuestos en la guía de usuario NiceLabel Automation.



#### NOTA

Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte la sección Acceso a recursos de red compartidos en la guía de usuario NiceLabel Automation.

El grupo **Contenido** establece detalles relacionados con el contenido del archivo.

- **Variable**: variable que almacena el contenido del archivo. Debe definirse al menos una variable (existente o recientemente creada).
- **Codificación**: tipo de codificación para los datos enviados. **Automática** define la codificación automáticamente. De ser necesario, seleccione el tipo de codificación preferida en la lista desplegable.



#### NOTA

La codificación no puede seleccionarse si los datos se leen desde una variable binaria. En este caso, la variable tendrá los datos como están.

El grupo **Reintentar en caso de error** define cómo debe continuar la ejecución de la acción si el archivo especificado se vuelve inaccesible.



#### SUGERENCIA

El módulo Automation Builder podría no estar en condiciones de acceder al archivo porque está bloqueado por otra aplicación. Si una aplicación continúa escribiendo datos en el archivo seleccionado y lo mantiene bloqueado en modo exclusivo, ninguna otra aplicación podrá abrirlo al mismo tiempo, ni siquiera para lectura. Otras causas posibles para los reintentos de la acción son: el archivo (aún) no existe, la carpeta (aún) no existe o el usuario del servicio no tiene privilegios para acceder al archivo.

- **Reintentos:** define la cantidad de reintentos para acceder al archivo. Si el valor está configurado en 0, no se realizará ningún reintento.
- **Intervalo de reintento:** intervalo de tiempo entre reintentos individuales en milisegundos.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.14.4. Eliminar archivo

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✗	✓	✓

Esta acción elimina un archivo seleccionado de un controlador.

El módulo NiceLabel Automation se ejecuta como servicio con una cuenta de usuario de Windows. Asegúrese de que la cuenta tenga permisos para eliminar el archivo en la carpeta especificada.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Archivo** establece los detalles relacionados al archivo.

- **Nombre de archivo:** nombre del archivo que se eliminará. El **Nombre de archivo** puede integrarse como parte del código. El **origen de datos** define en forma dinámica el **nombre del archivo** usando una variable existente o recientemente creada.

El nombre de ruta y archivo puede estar preprogramado, y el mismo archivo se usará cada vez.

Si solo se define un nombre de archivo sin ruta, se usa la carpeta con el archivo de configuración (.MISX) de NiceLabel Automation. Puede usar una referencia relativa para el nombre del archivo si se usa la carpeta con el archivo .MISX como carpeta raíz.

**Origen de datos** habilita el nombre de archivo variable. Seleccione o cree una variable que contenga la ruta de acceso o el nombre del archivo, o combine varias variables para crear el nombre de archivo.

Para obtener más información, consulte la sección Usar valores compuestos en la guía de usuario NiceLabel Automation.



#### NOTA

Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte la sección Acceso a recursos de red compartidos en la guía de usuario NiceLabel Automation.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

### 5.14.5. Ejecutar instrucción SQL

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise

Esta acción envía comandos SQL a un servidor SQL conectado y recopila los resultados. Utilice los comandos SELECCIONAR, INSERTAR, ACTUALIZAR y ELIMINAR.

Use la acción Ejecutar instrucción SQL para lograr estos dos objetivos:

- **Obtener datos adicionales de una base de datos:** En el módulo Automation Builder, el desencadenador recibirá datos para la impresión de etiquetas, pero no todos los valores requeridos. Por ejemplo, un desencadenador recibe valores para el `Product ID` y la `Description`, pero no para el `Price`. Debemos buscar el valor del `Precio` en la base de datos SQL.

#### Ejemplo de código SQL:

```
SELECT Price FROM Products
WHERE ID = :(Product ID)
```

El `ID` es el campo en la base de datos, `Product ID` es una variable definida en el desencadenador.



- **Actualizar o eliminar registros en una base de datos:** Después de imprimir una etiqueta, actualice el registro de la base de datos y envíe una señal al sistema de que ese registro en particular ya se ha procesado.

#### Ejemplo de código SQL:

Configure el valor del campo de la tabla `AlreadyPrinted` en `Verdadero` para el registro que se está procesando.

```
UPDATE Products
SET AlreadyPrinted = True
WHERE ID = :(Product ID)
```

O puede eliminar el registro actual de una base de datos, porque ya no es necesario.

```
DELETE FROM Products
WHERE ID = :(Product ID)
```

El `ID` es el campo en la base de datos, `Product ID` es una variable definida en el desencadenador.



#### NOTA

Para utilizar el valor de una variable dentro de una instrucción SQL, debe colocar dos puntos (:) delante de su nombre. Esto indica que sigue el nombre de una variable.



#### IMPORTANTE

Cuando cree su solución con una conexión de base de datos, use las variables de solicitud de sus etiquetas en lugar de los campos de la base de datos.

Para las variables de solicitud, use los mismos nombres que se definieron para los campos de la base de datos. Por ejemplo:

Campo de la base de datos: `food_products_1c.ProdCode`

Variable de solicitud de su etiqueta: `ProdCode`

NiceLabel identifica automáticamente las variables que corresponden a los campos de la base de datos.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Conexión de bases de datos** define la conexión de bases de datos que se utilizará para la instrucción.



### SUGERENCIA

Antes de que pueda enviar una sentencia SQL a una base de datos, configure la conexión con ella. Haga clic en el botón **Definir** y siga las instrucciones en pantalla. Puede conectarse a un origen de datos que puede controlarse con comandos SQL, de modo que no podrá utilizar archivos de texto (CSV) ni archivos Excel.

El grupo **Instrucción SQL** define la consulta o instrucción SQL que se ejecutará.



### SUGERENCIA

Las instrucciones del Lenguaje de manipulación de datos (DML) tienen permitido ejecutar consultas dentro de las tablas de bases de datos existentes.

Utilice instrucciones SQL estándares, como SELECCIONAR, INSERTAR, ELIMINAR y ACTUALIZAR, incluidas combinaciones, funciones y palabras clave. No se permiten las instrucciones en lenguaje DDL que se usan para crear bases de datos y tablas (CREAR BASE DE DATOS, CREAR TABLA), o para eliminarlas (ELIMINAR TABLA).

- **Prueba:** abre la sección **Vista previa de datos**. La opción Simular ejecución (seleccionada en forma predeterminada) prueba la ejecución de las instrucciones SQL. Haga clic en **Ejecutar** para iniciar la simulación.



### SUGERENCIA

La sección **Vista previa de datos** le permite probar la ejecución de su instrucción SQL con datos en directo. Para proteger los datos contra actualizaciones accidentales, asegúrese de que esté habilitada la opción **Simular ejecución**. Se ejecutarán las instrucciones INSERTAR, ELIMINAR y ACTUALIZAR. Esto le permite conocer los comentarios sobre cuántos registros se verán afectados; luego todas las transacciones se invierten.

Si utiliza variables de desencadenadores en la instrucción SQL, podrá ingresar sus valores para la ejecución de prueba.

- **Insertar origen de datos:** inserta variables predefinidas y recientemente creadas en una instrucción SQL.
- **Exportar/Importar:** habilita la exportación e importación de instrucciones SQL hacia/desde un archivo externo.
- **Modo de ejecución:** especifica el modo explícito de ejecución de la instrucción SQL.



## SUGERENCIA

En algunas consultas complejas de SQL, se vuelve cada vez más difícil determinar automáticamente cuál es la supuesta acción. Si la lógica integrada encuentra problemas para identificar su propósito, seleccione la acción principal manualmente.

- **Automática:** determina la acción automáticamente.
- **Devuelve un conjunto de registros (SELECCIONAR):** recibir el conjunto de datos con registros.
- **No devuelve un conjunto de registros (INSERTAR, ELIMINAR, ACTUALIZAR):** use esta opción si está ejecutando una consulta que no devuelve los registros. Inserte nuevos registros, elimine o actualice los registros existentes. El resultado es una respuesta de estado que indica el número de filas que se vieron afectadas por su consulta.
- **Tiempo de espera de ejecución:** le permite definir el tiempo de espera para enviar sus comandos al servidor SQL. Utilice el tiempo de espera de ejecución si está enviando varios comandos SQL consecutivos que requieren un tiempo de procesamiento más prolongado. Escriba la duración del tiempo de espera solicitado en segundos. Por defecto, la duración del tiempo de espera de ejecución es de 60 s. Si desea que el proveedor de su base de datos defina el tiempo de espera, escriba 0 s.

El grupo **Resultado** le permite configurar cómo se debe almacenar el resultado de la instrucción SQL, y definir la iteración de la acción.

- **Guardar datos en variable:** selecciona o crea una variable para almacenar el resultado de la instrucción SQL. Esta opción depende del **Modo de ejecución** seleccionado.
  - **Resultado de la instrucción SELECCIONAR.** Después de ejecutar una instrucción SELECCIONAR, el resultado es un conjunto de datos de registros. Recibirá el contenido de texto con formato CSV. La primera línea contiene los nombres de campo devueltos en un resultado. Las siguientes líneas contienen registros.



## NOTA

Para extraer los valores del conjunto de datos devuelto y para usarlos en otras acciones, defina y ejecute la acción Usar filtro de datos según los contenidos de esta variable (la acción está disponible en Automation Builder).

- **Resultado de las instrucciones INSERTAR, ELIMINAR y ACTUALIZAR.** Si utilizan las instrucciones INSERTAR, ELIMINAR y ACTUALIZAR, el resultado será un número que indica la cantidad de registros afectados en la tabla.
- **Iterar Para cada registro.** Si se habilita esta opción, NiceLabel agrega automáticamente una nueva acción **Para cada registro**. Para obtener más información sobre esta acción, consulte [el tema específico](#).



## NOTA

La asignación automática está habilitada. La acción Para cada registro no puede eliminarse.

Campo de la base de datos: `food_products_1c.ProdCode`

Variable de solicitud de su etiqueta: `ProdCode`

El grupo **Reintentar en caso de error** le permite configurar la acción para reintentar continuamente establecer la conexión con el servidor de una base de datos en caso de que el primer intento no tenga éxito. Si la acción no logra conectar en toda la cantidad de intentos definida, el error se eleva.

- **Reintentos:** especifica la cantidad de intentos para conectar con la base de datos.
- **Intervalo de reintento:** especifica la duración de tiempo entre los intentos individuales de reintento.

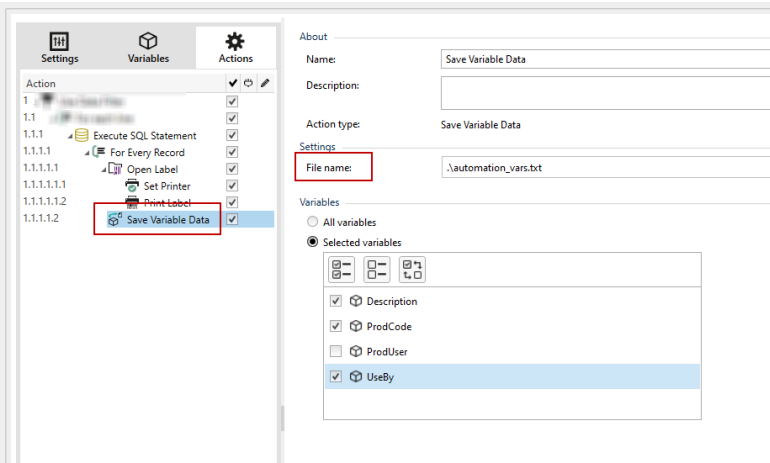
## Ejemplo

Desea imprimir etiquetas con datos de la base de datos `food_products_1c`, pero solo para los registros que tengan el valor de campo predefinido `UseBy`.

Usted define el valor `UseBy` con la variable `ProdUser`. En este caso, el valor de **ProdUser** es "3".

Una vez impresa cada etiqueta, NiceLabel escribe los valores de la base de datos en un archivo de texto en el disco. Use las siguientes acciones:

The screenshot shows the configuration window for an 'Execute SQL Statement' action. The left pane shows a tree view of actions, with 'Execute SQL Statement' selected. The right pane shows the configuration details for this action. The 'Name' is 'Execute SQL Statement'. The 'Action type' is 'Execute SQL Statement'. The 'Database Connection' is 'PEL...'. The 'SQL Statement' is 'SELECT \* FROM food\_products\_1c WHERE UseBy = :ProdUser'. The 'Execution mode' is 'Automatic'. The 'Execution timeout' is '60 s'. The 'Result' section has 'Iterate for Every Record' selected. Red boxes and arrows highlight the 'Execute SQL Statement' action in the left pane, the SQL statement text, and the 'Iterate for Every Record' option in the right pane.



Solo las etiquetas en las que el valor de UseBy es igual a "3" se imprimen, y sus valores se escriben en un archivo de texto:

```
"Description";"ProdCode";"UseBy"
"Choux pastry";"97700000011";"3"
"Croline";"97700000012";"3"
"Danish pastry";"97700000013";"3"
"Kouign-amann";"97700000022";"3"
"Muskazine";"97700000026";"3"
"Pineapple bun";"97700000029";"3"
"Schneeball";"97700000033";"3"
"Schuxen";"97700000034";"3"
"Tortell";"97700000041";"3"
```

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.14.6. Enviar datos a puerto TCP/IP

Esta acción envía los datos a un dispositivo externo, aceptando una conexión TCP/IP en un número de puerto predefinido.

**Enviar datos a puerto TCP/IP** establece la conexión con un dispositivo, envía los datos y termina la conexión. La conexión y la comunicación están regidas por el enlace que ocurre entre un cliente y un servidor cuando se inicia o finaliza una conexión TCP.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Configuración de conexión** establece los detalles de la conexión.

- **Responder al remitente:** Permite responder directamente al socket desde el cual se original los datos del desencadenador. Use esta opción para proporcionar comentarios acerca del proceso de impresión.



## NOTA

Esta opción está disponible en NiceLabel Automation.

Los prerequisites para la configuración de **Responder al remitente** son:

- La parte remota no cierra el canal de comunicación una vez que se entrega el mensaje.
- La acción **Enviar datos a puerto TCP/IP** se usa dentro del desencadenador **Servidor TCP/IP**.
- **No** configure el Evento de ejecución en el desencadenador del **servidor TCP/IP** como Cuando se **desconecta el cliente**.
- **Destino (dirección IP:puerto)**: dirección de destino y puerto del servidor TCP/IP. Codifique en forma rígida los parámetros de conexión y utilice el nombre de host o la dirección IP, o utilice los parámetros de conexión variables haciendo clic en la flecha derecha y seleccionando una variable predefinida. Para obtener más información, consulte el tema Combinación de valores en un objeto en la guía del usuario de NiceLabel Automation.

## Ejemplo

Si la variable `hostname` proporciona el nombre del servidor TCP/IP y la variable `puerto` proporciona el número de puerto, ingrese el siguiente parámetro para el destino:

```
[hostname] : [port]
```

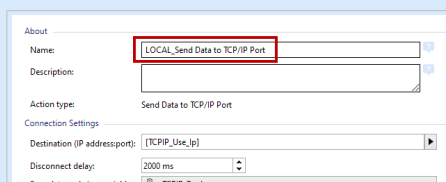
- **Retraso de desconexión**: prolonga la conexión con el socket de destino durante los intervalos de tiempo definidos después del envío de los datos. Ciertos dispositivos requieren más tiempo para procesar los datos. Inserte el retraso manualmente o haga clic en las flechas para aumentarlo o reducirlo.
- **Guardar respuesta de datos en una variable**: selecciona o crea una variable que almacena la respuesta del servidor. Los datos recibidos desde el servidor TCP/IP después de pasar el "retraso de desconexión" se almacenan en esta variable.



## NOTA

La acción **Enviar datos al puerto TCP/IP** se ejecuta en el Servidor de impresión web y no en el Cliente de impresión web.

Si utiliza la acción **Enviar datos al puerto TCP/IP** en NiceLabel Web Client y desea ejecutar localmente (sin pasar por el Servidor de impresión web), agregue el prefijo «LOCAL\_» al nombre de su acción.



El grupo **Contenido** define el contenido que se enviará a un servidor TCP/IP.



## SUGERENCIA

Utilice contenido fijo, una combinación de contenido fijo y variable, o contenido únicamente variable. Para ingresar contenido variable, haga clic en el botón con flecha que se encuentra a la derecha del área de datos e inserte una variable de la lista. Para obtener más información, consulte el tema **Combinación de valores en un objeto** en la guía del usuario de NiceLabel Automation.

- **Datos:** contenido de salida que se enviará.
- **Codificación:** tipo de codificación para los datos enviados. Automática define la codificación automáticamente. De ser necesario, seleccione el tipo de codificación preferida en la lista desplegable.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.



- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

### 5.14.7. Enviar datos a puerto de serie

Esta acción envía datos a un puerto de serie. Utilice esta acción para comunicarse con dispositivos con puertos de serie externos.



#### SUGERENCIA

Asegúrese de que la configuración del puerto coincida en ambos extremos; en la acción configurada y en el dispositivo del puerto de serie. El puerto de serie puede usarlo una sola aplicación de la máquina. Para usar satisfactoriamente el puerto desde esta acción, ninguna otra aplicación debe usar el puerto al mismo tiempo, ni siquiera un controlador de la impresora.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Puerto** define el puerto de serie.

- **Nombre de puerto:** nombre del puerto al cual se conecta un dispositivo externo. Puede ser un puerto COM de hardware o un puerto COM virtual.

El grupo **Configuración de puerto** define los parámetros adicionales de la conexión del puerto.

- **Bits por segundo:** tasa de velocidad utilizada por el dispositivo externo para comunicarse con la PC. El alias habitual que se utiliza con la configuración es "velocidad en baudios". Seleccione el valor del menú desplegable.
- **Bits de datos:** cantidad de datos en cada carácter. En casi todos los dispositivos más modernos, se utilizan 8 bits de datos. Seleccione el valor del menú desplegable.
- **Paridad:** método de detección de errores en una transmisión. La configuración de paridad más común es "ninguna", y la detección de errores está a cargo de un protocolo de comunicación (control de flujo). Seleccione el valor del menú desplegable.
- **Detener bits:** detiene los bits que se envían al final de cada carácter, con lo cual permite al hardware receptor de señal detectar el final de un carácter y resincronizarse con la secuencia de caracteres.

Los dispositivos electrónicos por lo general utilizan una detención única de bits. Seleccione el valor del menú desplegable.

- **Control de flujo:** el puerto de serie puede utilizar señales de interfaz para pausar y reanudar la transmisión de datos.

El grupo **Contenido** define el contenido que se enviará al puerto de serie.



### SUGERENCIA

Se permite contenido fijo, una combinación de contenido fijo y variable, o contenido únicamente variable. Para ingresar contenido variable, haga clic en el botón con flecha que se encuentra a la derecha del área de datos e inserte una variable de la lista. Para obtener más información, consulte el tema **Combinación de valores en un objeto** en la guía del usuario de NiceLabel Automation.

- **Datos:** contenido de salida que se enviará.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de

ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

### 5.14.8. Leer datos desde puerto de serie

Esta acción recopila los datos recibidos a través del puerto serial (RS-232) y lo guarda en una variable seleccionada. Utilice esta acción para comunicarse con dispositivos con puertos seriales externos.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Puerto** define el puerto de serie.

- **Nombre de puerto:** nombre del puerto al cual se conecta un dispositivo externo. Puede ser un puerto COM de hardware o un puerto COM virtual.

El grupo **Configuración de puerto** define los parámetros adicionales de la conexión del puerto.

- **Bits por segundo:** tasa de velocidad utilizada por el dispositivo externo para comunicarse con la PC. El alias habitual que se utiliza con la configuración es "velocidad en baudios".
- **Bits de datos:** especifica la cantidad de datos en cada carácter. En casi todos los dispositivos más modernos, se utilizan 8 bits de datos.
- **Paridad:** especifica el método de detección de errores en una transmisión. La configuración de paridad más común es "ninguna", y la detección de errores está a cargo de un protocolo de comunicación (control de flujo).
- **Detener bits:** detiene los bits que se envían al final de cada carácter, con lo cual permite al hardware receptor de señal detectar el final de un carácter y resincronizarse con la secuencia de caracteres. Los dispositivos electrónicos por lo general utilizan una detención única de bits.
- **Control de flujo:** el puerto de serie puede utilizar señales de interfaz para pausar y reanudar la transmisión de datos.

## Ejemplo

Es posible que los dispositivos lentos necesiten enlazarse con el puerto de serie para indicar que los datos deben pausarse mientras el dispositivo procesa los datos recibidos.

El grupo **Opciones** incluye los siguientes parámetros:

- **Retraso de lectura:** retraso opcional al leer datos desde el puerto de serie. Después del retraso, se lee todo el contenido del búfer del puerto de serie. Ingrese el retraso manualmente o haga clic en las flechas para aumentar o reducir el valor.
- **Enviar datos de inicialización:** especifica la cadena que se envía al puerto de serie seleccionado antes de que se lean los datos. Esta opción permite que la acción inicialice el dispositivo para que pueda suministrar los datos. La opción puede también utilizarse para enviar una pregunta específica al dispositivo y para recibir una respuesta específica. Haga clic en el botón de flecha para ingresar caracteres especiales.

El grupo **Extracción de datos** define cómo se extraen las partes definidas de los datos recibidos.

- **Posición inicial:** posición inicial para extracción de datos.
- **Posición final:** posición final para extracción de datos.

El grupo **Resultado** define una variable para el almacenamiento de datos.

- **Guardar resultado en variable:** selecciona o crea una variable para almacenar en los datos recibidos.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.14.9. Enviar datos a impresora

Esta acción envía datos a una impresora seleccionada. Úsela para enviar secuencias de impresora pregeneradas a cualquier impresora disponible.

El módulo NiceLabel Automation utiliza el controlador de la impresora instalada en modo indirecto solo para poder enviar datos al puerto de destino, por ejemplo, un puerto USB, TCP/IP, COM o LPT, al cual está conectada la impresora.



## NOTA

Escenario posible. Los datos recibidos por el desencadenador deben imprimirse en la misma impresora de red en una plantilla de etiqueta diferente (archivos de etiquetas .NLBL). La impresora puede aceptar datos de varias estaciones de trabajo y, generalmente, imprimirá los trabajos en el orden en que los recibió. El módulo Automation Builder enviará cada plantilla de etiqueta en un trabajo de impresión separado, de modo que otra estación de trabajo pueda insertar sus trabajos entre los creados en nuestro módulo Automation Builder. En lugar de enviar cada trabajo en forma separada a la impresora, puede combinar todos los trabajos juntos (usando la acción [Redireccionar impresión a archivo](#)) y luego enviar un solo gran trabajo a la impresora.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.

- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Impresora** selecciona la impresora.

- **Nombre de impresora:** nombre de la impresora a la cual se enviarán los datos. Seleccione la impresora desde la lista desplegable de controladores de impresora instalados en forma local, ingrese un nombre de impresora personalizado y defínalo dinámicamente usando una variable nueva o creada recientemente.

El grupo **Origen de datos** define el contenido que se enviará a la impresora.

- **Usar datos recibidos por el desencadenador:** datos usados recibidos por el desencadenador. En este caso, quiere que la secuencia de impresora recibida se use como entrada para el filtro. Su objetivo es redireccionarlo a una impresora sin ninguna modificación. El mismo resultado puede archivarse habilitando la variable interna `DataFileName` y usando el contenido del archivo al que se refiere. Para obtener más información, consulte la sección Usar valores compuestos en la guía de usuario NiceLabel Automation.
- **Nombre de archivo:** ruta de acceso y nombre de archivo que contiene la secuencia de una impresora. El contenido del archivo específico se envía a una impresora. Seleccione **Origen de datos** para definir el nombre de archivo en forma dinámica usando un valor de variable.
- **Variable:** variable (nueva o existente) que almacena la secuencia de la impresora.
- **Personalizar:** define el contenido personalizado que se enviará a una impresora. Se permite contenido fijo, una combinación de contenido fijo y variable, o contenido únicamente variable. Para ingresar contenido variable, haga clic en el botón con flecha que se encuentra a la derecha del área de datos e inserte una variable de la lista. Para obtener más información, consulte el tema Combinación de valores en un objeto en la guía del usuario de NiceLabel 10.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (`verdadero` o `falso`). Si el resultado de la expresión es `verdadero`, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.14.10. Solicitud de HTTP

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✗	✗	✓

Esta acción envía datos al servidor web de destino utilizando el método HTTP seleccionado. Se permiten los esquemas HTTP y HTTPS URL.

HTTP funciona como un protocolo de solicitud-respuesta en el modelo computacional cliente-servidor. En esta acción, NiceLabel 10 actúa como cliente que se comunica con un servidor remoto. Esta acción envía un mensaje de solicitud de HTTP seleccionado a un servidor. El servidor devuelve un mensaje de respuesta que puede contener información de estado de finalización sobre la solicitud y puede también incluir el contenido solicitado en el cuerpo del mensaje.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Configuración de conexión** establece los parámetros de la conexión.



### NOTA

Esta acción admite el Protocolo de Internet versión 6 (IPv6).

- **Destino:** dirección, puerto y destino (ruta directa) del servidor web.



### SUGERENCIA

Si el servidor web se ejecuta en el puerto predeterminado 80, omita el número de puerto. Integre como parte del código los parámetros de conexión y utilice un nombre de host fijo o la dirección IP. Utilice un valor de variable para definir esta opción en forma dinámica. Para obtener más información, consulte la sección Usar valores compuestos en la guía de usuario NiceLabel Automation.

### Ejemplo

Si la variable `hostname` proporciona el nombre del servidor web y la variable `port` proporciona el número de puerto, puede ingresar lo siguiente para el destino:

```
[hostname] : [port]
```

- **Método de solicitud:** métodos de solicitud disponibles.
- **Tiempo de espera:** duración del tiempo de espera (en ms) durante el cual debe establecerse la conexión al servidor y recibirse respuesta.
- **Guardar respuesta de estado en una variable:** variable para almacenar el código de estado recibido desde el servidor.



### SUGERENCIA

El código de estado en el rango 2XX es un código exitoso. La respuesta "OK" más común es código 200. Los códigos 5XX son errores de servidor.

- **Guardar respuesta de datos en una variable:** variable para almacenar los datos recibidos desde el servidor.

El grupo **Autenticación** le permite asegurar la conexión al servidor web.

- **Habilitar autenticación básica:** le permite ingresar las credenciales necesarias para conectarse al servidor web. El nombre de usuario y la contraseña pueden ser fijos o proporcionarse usando una variable.





## NOTA

La autenticación HTTP básica (BA) utiliza encabezados HTTP estáticos convencionales. El mecanismo BA no brinda protección de la confidencialidad de las credenciales transmitidas. Solo se codifican con Base64 en tránsito, pero no se cifran ni reciben algoritmo hash de ningún modo. La autenticación básica debe utilizarse sobre HTTPS.

- **Mostrar contraseña:** desenmascara los caracteres de la contraseña.



## NOTA

La acción **Solicitud de HTTP** se ejecuta en el Servidor de impresión web y no en el Cliente de impresión web.

Si utiliza la acción **Solicitud de HTTP** en NiceLabel Web Client y desea ejecutar localmente (sin pasar por el Servidor de impresión web), agregue el prefijo «LOCAL\_» al nombre de su acción.

**Contenido** define el contenido que se enviará a un servidor web.

- **Datos:** contenido de salida que se enviará. Se permite contenido fijo, una combinación de contenido fijo y variable, o contenido únicamente variable. Para ingresar contenido variable, haga clic en el botón con flecha que se encuentra a la derecha del área de datos e inserte una variable de la lista. Para obtener más información, consulte el tema Combinación de valores en un objeto en la guía del usuario de NiceLabel 10.
- **Codificación:** tipo de codificación para los datos enviados.



## SUGERENCIA

**Automática** define la codificación automáticamente. De ser necesario, seleccione el tipo de codificación preferida en la lista desplegable.

- **Tipo:** propiedad del tipo de contenido del mensaje HTTP. Si no se selecciona ningún tipo, se utiliza el tipo predeterminado `application/x-www-form-urlencoded`. Si no se enumera un tipo apropiado, defina uno personalizado o establezca una variable que lo defina en forma dinámica.

Ciertos servidores HTTP requieren **Encabezados HTTP adicionales** (especialmente para servicios REST).



## NOTA

La acción **Solicitud de HTTP** ya utiliza una clave de suscripción incorporada, por lo que no es necesario ingresar el valor **Ocp-Apim-Subscription-Key** en **Encabezados HTTP adicionales**.

Debe proporcionar el valor **Ocp-Apim-Subscription-Key** si realiza una llamada a la API de la nube desde una aplicación que no es de NiceLabel.

- **Encabezados adicionales:** encabezados integrados como parte del código u obtenidos a partir de valores de variable. Para acceder a las variables, haga clic en el botón de flecha pequeño que se encuentra a la derecha del área de texto. Para obtener más información, consulte el tema **Combinación de valores en un objeto** en la guía del usuario de NiceLabel 10.

Algunos servidores HTTP (especialmente para servicios REST) requieren que se incluyan encabezados HTTP personalizados en el mensaje. Esta sección le permite proporcionar el encabezado de HTTP requerido.

Los encabezados HTTP deben ingresarse con la siguiente sintaxis:

```
header field name: header field value
```

Por ejemplo, para utilizar los nombres de campo de encabezado **Aceptar**, **Usuario-Agente** y **Contenido-Tipo**, puede utilizar la siguiente sintaxis:

```
Accept: application/json; charset=utf-8
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/31.0.1650.63 Safari/537.36
Content-Type: application/json; charset=UTF-8
```

Puede integrar los nombres de campo del encabezado, o puede obtener los valores a partir de las variables de desencadenadores. Utilice tantos campos de encabezado personalizados como desee, siempre y cuando se asegure de colocar cada campo de encabezado en una línea nueva.



## NOTA

Los encabezados HTTP ingresados sustituirán los encabezados ya definidos en cualquier otro lado de las propiedades de las acciones, como **Contenido-Tipo**.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.

- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.14.11. Servicio Web

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✗	✗	✓

Un servicio Web es un método de comunicación entre dos dispositivos electrónicos o instancias de software. Un servicio Web se define como un estándar de intercambio de datos. Utiliza formato XML para etiquetar los datos; el protocolo SOAP se usa para transferir los datos y se usa lenguaje WSDL para describir los servicios disponibles.

Esta acción establece la conexión con un servicio Web remoto y ejecuta los métodos allí. Los métodos se pueden describir como acciones que se publican en el Servicio Web. La acción envía los valores de entrada al método seleccionado en el servicio Web remoto, recopila el resultado y lo guarda en las variables seleccionadas.

Después de importar el WSDL y agregar una referencia al Servicio Web, sus métodos se incluirán en el cuadro combinado Método.



## NOTA

Puede transferir tipos simples al Servicio Web, como cadenas, enteros, booleanos, pero no tipos complejos. El WSDL debe contener únicamente un enlace.



## NOTA

Debe imprimir etiquetas de productos. Su desencadenador recibirá únicamente parte de los datos necesarios. Por ejemplo, el desencadenador recibe el valor para las variables **Product ID** y la **Description**, pero no para **Price**. El precio está disponible en una base de datos separada, accesible desde la llamada del servicio Web. El servicio Web define la función usando una definición del WSDL. Por ejemplo, la entrada de la función es **Product ID** y su salida es **Price**. La acción del servicio Web envía **Product ID** al servicio Web. Ejecuta y hace una búsqueda interna en su base de datos y proporciona el **Price** correspondiente como resultado. La acción guarda el resultado en una variable, la cual se puede usar en la etiqueta.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Definición de servicio Web** incluye los siguientes parámetros:



## NOTA

Esta acción admite el Protocolo de Internet versión 6 (IPv6).

- **WSDL:** ubicación de definición del WSDL.  
El WSDL generalmente es provisto por el propio servicio Web. Comúnmente, debe ingresar el vínculo al WSDL y hacer clic en el botón **Importar** para leer la definición. Si tiene problemas para obtener el WSDL del recurso con conexión, guarde el WSDL en un archivo e ingrese la ruta de acceso con el nombre del archivo para cargar métodos desde allí. NiceLabel 10 automáticamente detecta si el Servicio Web remoto utiliza la sintaxis de RPC o documentos y si se comunica correctamente o no.
- **Dirección:** dirección en la que se publica el Servicio Web.  
En principio, esta información se toma del WSDL, pero puede actualizarse antes de ejecutar la acción. Es útil para entornos de desarrollo dividido, prueba y producción, en los que se utiliza la misma lista de acciones, pero con diferentes nombres de servidores donde se ejecutan los Servidores Web. Se permite contenido fijo, una combinación de contenido fijo y variable, o contenido únicamente variable. Para ingresar contenido variable, haga clic en el botón con flecha que se encuentra a la

derecha del área de datos e inserte una variable de la lista. Para obtener más información, consulte el tema Combinación de valores en un objeto en la guía del usuario de NiceLabel 10.

- **Método:** métodos (funciones) que están disponibles en un servicio Web seleccionado. La lista la llena automáticamente la definición del WSDL.
- **Parámetros:** variables de entrada y salida para el método seleccionado (función).  
Los parámetros de entrada esperan una entrada. Por motivos de prueba y resolución de problemas, puede ingresar un valor fijo y ver el resultado de la vista previa en la pantalla. Generalmente, seleccionará una variable para el parámetro de entrada. El valor de esa variable se utilizará como parámetro de entrada. El parámetro de salida proporciona un resultado desde la función. Debe seleccionar la variable que almacenará el resultado.
- **Tiempo de espera:** tiempo de espera después del cual se establece la conexión al servidor.



## NOTA

La acción de **Servicio web** se ejecuta en el Servidor de impresión web y no en el Cliente de impresión web.

Si utiliza la acción de **Servicio web** en NiceLabel Web Client y desea ejecutar localmente (sin pasar por el Servidor de impresión web), agregue el prefijo «LOCAL\_» al nombre de su acción.

The screenshot shows a configuration window for a web service action. The 'Name' field is highlighted with a red box and contains the text 'LOCAL\_Web Service'. Other fields include 'Description', 'Action type: Web Service', 'WSDL: http://192.168.148.18:56', 'Address: [WebService\_Host]', 'Method: ExecuteTriggerWithResponse', and 'Timeout: 100000 ms'.

La **Autenticación** habilita la autenticación básica del usuario. Esta opción define las credenciales del usuario necesarias para establecer una llamada de salida a un servidor Web remoto.

- **Habilitar la autenticación básica:** permite definir el **Nombre de usuario** y la **Contraseña**, que se pueden escribir en forma manual o definirse con los valores variables. Seleccione los **Orígenes de datos** para seleccionar o crear las variables.
- **Mostrar contraseña:** muestra los caracteres enmascarados del **Nombre de usuario** y la **Contraseña**. Encontrará detalles acerca de los problemas de seguridad en la sección Asegurar el acceso a sus desencadenadores en la guía del usuario de NiceLabel Automation.

El campo **Vista previa de datos** le permite ejecutar una prueba del servicio Web.

- El botón **Ejecutar** ejecuta una llamada del servicio Web.  
Envía los valores de los parámetros de entrada al servicio Web y proporciona el resultado en el parámetro de salida. Use esta funcionalidad para probar la ejecución de un servicio Web. Puede

ingresar valores para los parámetros de entrada y ver el resultado en pantalla. Una vez satisfecho con la ejecución, reemplace el valor fijo ingresado para el parámetro de entrada con una variable de la lista.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

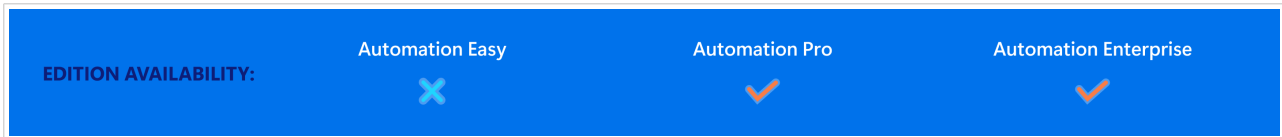
### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.15. Otros

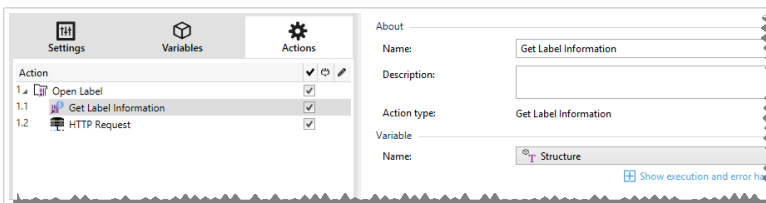
### 5.15.1. Obtener información de etiqueta



Esta acción devuelve información estructural sobre el archivo de etiqueta asociado. La acción proporciona información sobre las dimensiones de la etiqueta, el controlador de la impresora y todas las variables de la etiqueta y sus propiedades principales.

La acción Obtener información de etiqueta devuelve la información original guardada en el archivo de la etiqueta. Además, proporciona información una vez que se ha simulado el proceso de impresión. La simulación garantiza que todas las variables de etiquetas obtengan el valor que tendrían durante una impresión normal. Asimismo, la información de la altura de la etiqueta proporciona las dimensiones correctas si la etiqueta está definida como de altura variable (en cuyo caso el tamaño de la etiqueta depende de la cantidad de datos que se imprimirán). La acción devolverá las dimensiones para un tamaño de etiqueta, no para un tamaño de página.

La acción guarda información de la estructura de la etiqueta en una variable seleccionada. Puede enviar los datos de vuelta al sistema utilizando la acción Solicitud HTTP (o una acción de conectividad de datos de salida similar), o en la respuesta del desencadenador, si este es bidireccional.



#### NOTA

Esta acción debe anidarse debajo de la acción [Abrir etiqueta](#).

El grupo **Variable** selecciona o crea una variable que almacena la información estructural acerca de una etiqueta.

- **Nombre:** especifica el nombre de la variable. Selecciona o crea una variable que almacena la información de la etiqueta en formato XML.
  - Si desea utilizar la información de XML dentro de este desencadenador, puede definir el Filtro XML y ejecutarlo con la acción Usar filtro de datos (Automation Builder solamente).
  - Si desea devolver los datos XML como respuesta en su desencadenador HTTP o de Servicio Web, utilice esta variable directamente en el campo **Datos de respuesta** de la página de configuración del desencadenador.

- Si desea guardar los datos XML en un archivo, utilice la acción [Guardar datos en el archivo](#).

El grupo **Configuración adicional** le permite habilitar el uso de valores provisionales.

- **Usar valores provisionales:** reemplaza valores de orígenes de datos faltantes con valores provisionales.



### SUGERENCIA

Consulte la sección Variable en NiceLabel 10 la guía del usuario de Designer para conocer una descripción detallada de los valores provisionales.

### Información de etiqueta XML de muestra

La siguiente muestra presenta una vista estructural de los elementos y sus atributos tal cual se devuelven.

```
<?xml version="1.0" encoding="UTF-8"?>
<Label>
  <Original>
    <Width>25000</Width>
    <Height>179670</Height>
    <PrinterName>QLS 3001 Xe</Printer>
  </Original>
  <Current>
    <Width>25000</Width>
    <Height>15120</Height>
    <PrinterName>QLS 3001 Xe</Printer>
  </Current>
  <Variables>
    <Variable>
      <Name>barcode</Name>
      <Description></Description>
      <DefaultValue></DefaultValue>
      <Format>All</Format>
      <CurrentValue></CurrentValue>
      <IncrementType>None</IncrementType>
      <IncrementStep>0</IncrementStep>
      <IncrementCount>0</IncrementCount>
      <Length>100</Length>
    </Variable>
  </Variables>
</Format>
```

### Especificación de información de etiqueta XML



Esta sección contiene una descripción de la estructura del archivo XML según la devuelve la acción Obtener información de etiqueta.



## NOTA

Todos los valores de medidas se expresan en unidades 1/1000 mm. Por ejemplo, un ancho de 25000 es 25 mm.

- **<Label>**: este es un elemento de raíz.
- **<Original>**: especifica las dimensiones de la etiqueta y el nombre de la impresora como están guardados en el archivo de etiqueta.
  - **Ancho**: este elemento contiene el ancho original de la etiqueta.
  - **Altura**: este elemento contiene la altura de la etiqueta original.
  - **PrinterName**: Este elemento contiene el nombre de la impresora para la cual se ha creado la etiqueta.
- **Actual**: especifica las dimensiones de la etiqueta y el nombre de la impresora después de que se completa la impresión simulada.
  - **Ancho**: este elemento contiene el ancho real de la etiqueta.
  - **Altura**: este elemento contiene la altura real de la etiqueta. Si una etiqueta está definida como de altura variable, puede crecer junto con los objetos de la etiqueta. Por ejemplo, el tamaño de los objetos RTF y Cuadro de texto puede crecer en dirección vertical y hacer que la etiqueta también se expanda.
  - **PrinterName**: este elemento contiene el nombre de la impresora que se utilizará para imprimir.

## Ejemplo

Se usará una impresora distinta de la original si el controlador de impresora original no está instalado en esta computadora o si la impresora se ha cambiado usando la acción [Establecer impresora](#).

- **<Variables>** y **<Variable>**: el elemento `variables` contiene la lista de todas las variables de etiquetas de solicitud, cada una definida en un elemento `variable` separado. Las variables de solicitud son aquellas que se incluyen en el cuadro de diálogo de impresión cuando se imprimen etiquetas desde el NiceLabel 10 . Si no hay variables de solicitud definidas en la etiqueta, el elemento `variables` estará vacío.
  - **Nombre**: contiene el nombre de variable.
  - **Descripción**: contiene la descripción de variable.
  - **DefaultValue**: contiene el valor predeterminado según lo definido para la variable durante el proceso de diseño.

- **Formato:** contiene el tipo aceptable de contenido variable (caracteres).
- **IsPrompted:** contiene información acerca de si la variable es solicitada o no al momento de imprimir.
- **PromptText:** contiene texto que el solicita al usuario la entrada de un valor.
- **CurrentValue:** contiene el valor real que se usa para imprimir.
- **IncrementType:** contiene información, si la variable se definió o no como contador. Si se definió como contador, informa qué tipo de contador es.
- **IncrementStep:** contiene información sobre el paso del contador. El valor del contador aumenta o disminuye según este valor en la siguiente etiqueta.
- **IncrementCount:** contiene información acerca del punto de aumentar/disminuir el valor de contador. Por lo general, el contador cambia de valor en cada etiqueta, pero eso puede modificarse.
- **Longitud:** contiene el número máximo de caracteres almacenados en una variable.
- **IsPickListEnabled:** contiene información acerca de si el usuario selecciona o no valores variables de una lista de selección.
- **PickListValues:** contiene los valores reales (seleccionables) de la lista de selección.

#### Definición de esquema XML (XSD) para especificación de etiqueta XML

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema id="Format" xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Label">
    <xs:complexType>
      <xs:all>
        <xs:element name="Original">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Width" type="xs:decimal"
minOccurs="1" />
              <xs:element name="Height" type="xs:decimal"
minOccurs="1" />
              <xs:element name="PrinterName" type="xs:string"
minOccurs="1" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element name="Current">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Width" type="xs:decimal"
minOccurs="1" />
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        <xs:element name="Height" type="xs:decimal"
minOccurs="1" />
        <xs:element name="PrinterName" type="xs:string"
minOccurs="1" />
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Variables">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Variable" minOccurs="0"
maxOccurs="unbounded">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element name="Name"
type="xs:string" minOccurs="1" />
                        <xs:element name="Description"
type="xs:string" minOccurs="1" />
                        <xs:element name="DefaultValue"
type="xs:string" minOccurs="1" />
                        <xs:element name="Format"
type="xs:string" minOccurs="1" />
                        <xs:element name="CurrentValue"
type="xs:string" minOccurs="1" />
                        <xs:element name="IncrementType"
type="xs:string" minOccurs="1" />
                        <xs:element name="IncrementStep"
type="xs:integer" minOccurs="1" />
                        <xs:element name="IncrementCount"
type="xs:integer" minOccurs="1" />
                        <xs:element name="Length"
type="xs:string" minOccurs="1" />
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:all>
</xs:complexType>
</xs:element>
</xs:schema>

```

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

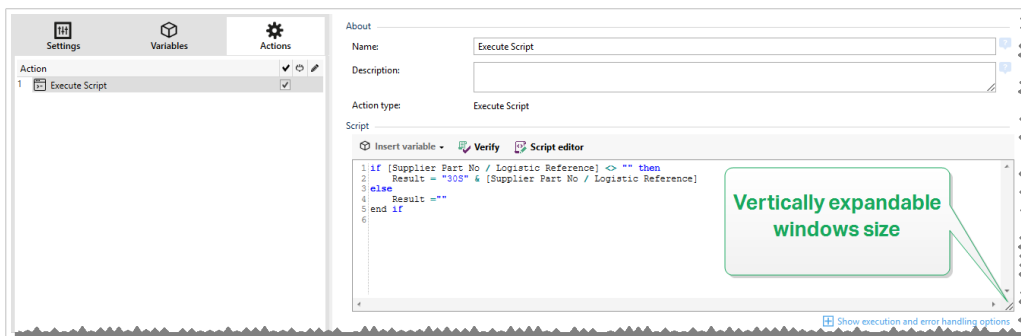
- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables **ActionLastErrorId** y **ActionLastErrorDesc**.

## 5.15.2. Ejecutar secuencia de comandos

Esta acción mejora la funcionalidad de software utilizando las secuencias de comandos personalizadas VBScript o Python. Utilice esta acción si las acciones integradas no cumplen con los requisitos de manipulación de datos existentes.

Las secuencias de comandos pueden incluir las variables de los desencadenadores, tanto las internas como las que usted define o importa desde etiquetas.

Asegúrese también de que la cuenta de Windows con la que se utiliza el servicio tenga privilegios para ejecutar los comandos de la secuencia.



## NOTA

El tipo de secuencia de comandos se configura para cada desencadenador en sus propiedades. Todas las acciones Ejecutar secuencia de comandos dentro de un desencadenador deben ser del mismo tipo.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El editor de **Secuencias de comandos** ofrece las siguientes funciones:

- **Insertar origen de datos:** inserta una variable existente o creada recientemente en la secuencia de comandos.
- **Verificar:** valida la sintaxis de la secuencia de comandos ingresada.
- **Editor de Secuencias de comandos:** abre el editor, que facilita la creación de secuencias de comandos y la hace más eficiente.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

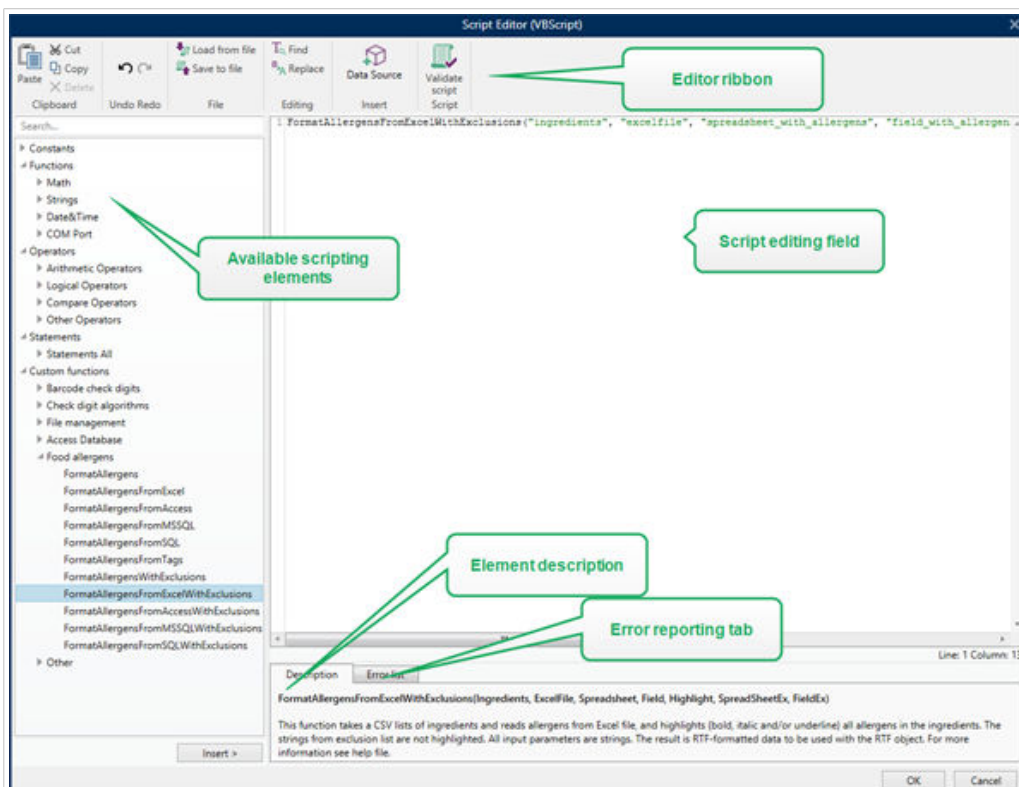
## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables **ActionLastErrorId** y **ActionLastErrorDesc**.

### 5.15.2.1. Editor de secuencias de comandos

NiceLabel 10 proporciona un editor de secuencias de comandos que facilita la creación de secuencias de Python o VBScript, no presenta errores y es más rápido.



La selección de los idiomas de secuencias que deben usarse en el Editor de secuencias de comandos difiere entre NiceLabel Designer Pro y Automation Builder:

- En Designer, haga doble clic en la superficie de diseño del formulario para abrir Propiedades del formulario > Configuración adicional > Idioma de secuencias de comandos.
- En Automation Builder, vaya a Elementos de configuración > haga clic en Editar para abrir las propiedades del desencadenador > Configuración > Otras > Secuencia de comandos.

NiceLabel 10 usa la variante .NET de Python llamada IronPython. Funciona como una implementación totalmente compatible del idioma de la secuencia de comandos de Python y también es compatible con los métodos .NET.

La banda de opciones del Editor incluye los comandos utilizados comúnmente, que se distribuyen a múltiples grupos de funciones.

- Grupo Portapapeles ofrece los comandos Cortar, Copiar, Pegar y Eliminar.
- Grupo Deshacer permite deshacer y repetir acciones de edición de secuencias de comandos.
- El grupo Archivo permite cargar y guardar secuencias de comando en un archivo.
  - Cargar desde archivo: carga una secuencia de comandos desde un archivo textual externo previamente guardado.
  - Guardar en el archivo: almacena una secuencia de comandos editada actualmente en un archivo de texto.
- Grupo Edición: permite buscar y reemplazar cadenas en una secuencia de comandos.
  - Buscar: busca la cadena ingresada en la secuencia de comandos.
  - Reemplazar: reemplaza la cadena en la secuencia de comandos.
- El grupo Insertar: el comando Origen de datos inserta orígenes de datos existentes o definidos recientemente en la secuencia de comandos.
- Grupo Secuencia de comandos: el comando Validar secuencia de comandos valida la sintaxis de la secuencia de comandos ingresada.

Los Elementos de secuencias de comandos disponibles contienen todos los elementos de la secuencia de comandos disponibles que se pueden utilizar al crear una secuencia de comandos. Haga doble clic en el elemento o clic en el botón Insertar para insertar el elemento en la posición del cursor en la secuencia de comandos.

La Descripción de elementos brinda información básica sobre el elemento de la secuencia de comandos insertado.

La Lista de errores presenta los errores que se informaron después de ejecutar el comando Validar secuencia de comandos.

### 5.15.3. Mensaje

Utilice la acción de mensaje para escribir cadenas personalizadas (por ejemplo, mensajes personalizados de advertencia, valores de variables y comentarios). La acción de mensaje crea entradas personalizadas en los archivos de registro del Automation Manager. Los archivos de registro de automatización contienen información generada por la aplicación, las advertencias y las descripciones de errores. Utilice los registros de mensaje para realizar un seguimiento de las variables de mensajes durante la configuración, la resolución de problemas y la depuración.

Para configurar acciones de mensaje, haga lo siguiente:

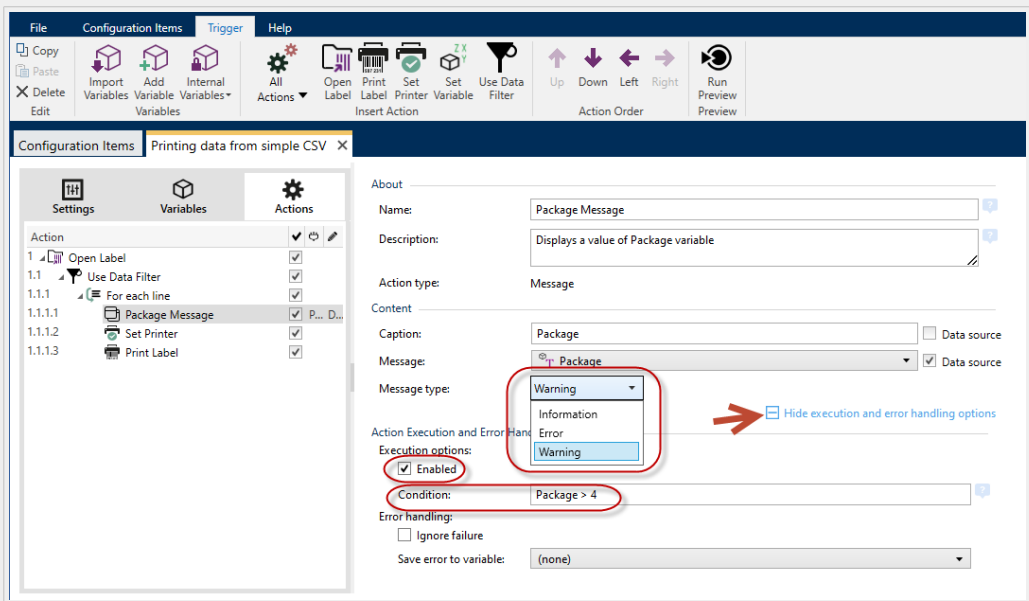
1. Vaya a **Todas las acciones** y seleccione **Mensaje** en el menú desplegable de **Acción**.
2. Cambie el nombre de la acción e ingrese una descripción.
3. Configure el **contenido** del mensaje: **título, mensaje y tipo de mensaje**.  
Algunos **tipos de mensajes** son:
  - **Información**
  - **Error**
  - **Advertencia**
4. Abra **Mostrar opciones de ejecución de acciones y tratamiento de errores** para establecer las condiciones que deben darse para mostrar los mensajes, ignorar los fallos y guardar los errores de automatización en las variables.

El Automation Manager de NiceLabel muestra mensajes con distintos colores (como errores en rojo y advertencias en naranja) en el panel de registro de automatización.

#### Ejemplo:

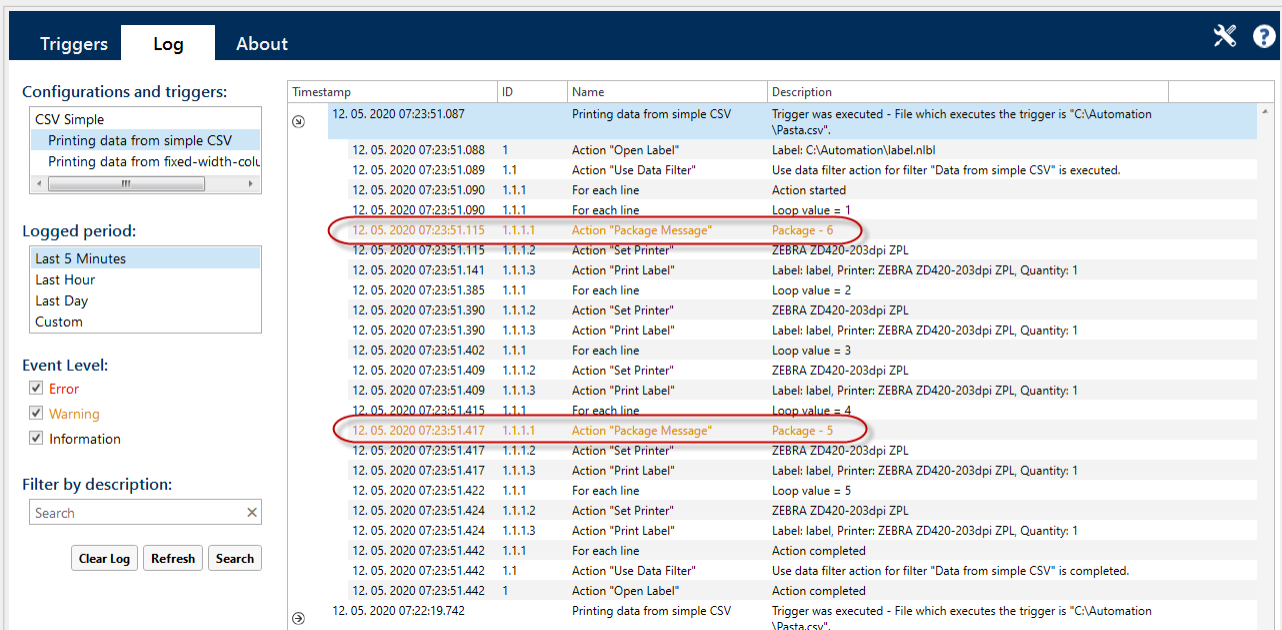
Imprime etiquetas de pasta con automatización. El desencadenador recibe valores de variables de los archivos CSV generados en ERP. Cuando el valor de la variable «Paquete» es mayor a 4, el registro de automatización crea una advertencia.





Configuración de acciones de **mensaje**.

El resultado en Automation Manager se ve de la siguiente manera:



## NOTA

Cuando configura la gravedad de mensajes como «error», los desencadenadores no pasan al estado de error. La impresión sigue siendo posible.

Use registros de mensaje para:

- configurar la resolución de problemas
- depurar las soluciones

- hacer un seguimiento de los valores de las variables seleccionadas

mostrar los mensajes personalizados de error y de advertencia

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables **ActionLastErrorId** y **ActionLastErrorDesc**.

### 5.15.4. Verificar licencia

Esta acción lee la licencia activada y ejecuta las acciones anidadas debajo de esta acción únicamente si se utiliza cierto tipo de licencia.



## SUGERENCIA

La acción Verificar licencia proporciona protección para la configuración del desencadenador contra la ejecución en máquinas no autorizadas.



## NOTA

La clave de la licencia que activa el software también puede codificar la ID de solución. Es un número único que identifica al proveedor de la solución que vendió la licencia NiceLabel 10.

Si el ID de la solución coincide con el ID de la solución codificado en la licencia, la máquina de destino puede ejecutar las acciones anidadas, limitando efectivamente la ejecución de licencias vendidas por el proveedor de la solución.

Los desencadenadores pueden cifrarse más y bloquearse de modo que solo los usuarios autorizados puedan abrir la configuración. Para obtener más información, consulte la sección Proteger configuración de desencadenadores en la guía de usuario de NiceLabel Automation.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Información de licencia** le permite seleccionar el ID de la licencia.

- El **ID de la licencia:** define el número de licencias que pueden ejecutar las acciones anidadas.
  - Si el valor ingresado no es el ID de la licencia codificado en la licencia, las acciones anidadas no se ejecutan.
  - Si el valor ingresado es 0, las acciones se ejecutan si se encuentra una licencia válida.



## NOTA

El UID de Socio Digital también puede usarse como ID de licencia. Esta opción está disponible para miembros del [Programa de Socio Digital de NiceLabel](#).

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

### 5.15.5. Intentar

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✗	✗	✓

Esta acción le permite:

- Monitorear los errores mientras se ejecutan las acciones.
- Ejecutar un conjunto de acciones alternativas, si se produce un error.

La acción Intentar crea los marcadores de posición **Hacer** y **Al ocurrir un error** para las acciones. Todas las acciones que deban ejecutarse cuando los desencadenadores se activan, deben colocarse dentro del marcador de posición **Hacer**. Si no se detecta ningún error al ejecutar acciones desde el marcador de posición **Hacer**, estas son las únicas acciones que se ejecutan. No obstante, si ocurre un error, la ejecución de acciones del marcador de posición **Hacer** se detiene y la ejecución se transfiere a las acciones del marcador de posición **Al ocurrir un error**.



## NOTA

Debe habilitar la [Impresión sincronizada](#) para detectar errores con **Al ocurrir un error**.

		Settings	Variables	Actions
Action		✓	🔄	✎
1	For loop	✓		
2	Try	✓		
2.1	Do	✓		
2.1.1	Open Label	✓		
2.1.1.1	Set Printer	✓		
2.1.1.2	Print Label	✓		
2.2	On Error	✓		
2.2.1	HTTP Request	✓		
3	Read Data from File	✓		

## Ejemplo

Si falla alguna de las acciones del marcador de posición Hacer, se detiene la ejecución de la acción y se reanudan las acciones del marcador de posición Al ocurrir un error. Si Intentar se colocara independientemente, terminaría la ejecución del desencadenador. En este caso, Intentar está anidada debajo de la acción Para bucle. Normalmente, todo error en el marcador de posición Hacer también detendría la ejecución de la acción Para bucle, incluso si todavía hay más pasos hasta completar la acción de bucle. En este caso, Guardar datos en archivo tampoco se ejecutará. En forma predeterminada, cada error interrumpe el procesamiento total de los desencadenadores.

Sin embargo, puede continuar con la ejecución de la siguiente iteración en la acción Para bucle. Para que esto suceda, habilite la opción Ignorar error en la acción Intentar. Si los datos del paso actual en Para bucle producen un error en el marcador de posición Hacer, se ejecutan las acciones de Al ocurrir un error. A continuación, se ejecuta Guardar datos en archivo en nivel 2 y luego la acción Para bucle continúa su ejecución en la siguiente iteración.



## SUGERENCIA

Esta acción brinda la fácil detección de errores y ejecución de "comentarios" e "informes". Por ejemplo, si ocurre un error durante el procesamiento del desencadenador, puede enviar una advertencia. Para obtener más información, consulte la sección Comentarios sobre el estado del trabajo de impresión en la guía de usuario NiceLabel Automation.



## NOTA

**¡Importante!** La acción **Intentar** proporciona los resultados previstos con acciones asincrónicas. Si el bucle Intentar incluye la acción [Imprimir etiqueta](#) que falla, la ejecución de la acción aún completa el bucle Intentar y no cambia a las acciones **Al ocurrir un error** como se espera. El resultado de no cambiar a las acciones **Al ocurrir un error** es la acción Imprimir etiqueta que se ejecuta de modo sincronizado de forma predeterminada. Para evitarlo, asegúrese de que esté activada la impresión supervisada. Vaya a la configuración del desencadenador > **Otro** > **Comentarios del motor de impresión** y habilite **Impresión supervisada**.

Lea más sobre la impresión supervisada en la sección [Modo de impresión sincronizada](#).

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

### 5.15.6. Transformar XML

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	X	X	✓

Esta acción transforma el documento XML en otro documento usando las reglas de transformación provistas. Las reglas debe proporcionarlas la definición .XSLT en un archivo, u otro origen de variable.

La acción permite convertir documentos XML complejos en documentos XML con una estructura más fácil de administrar. XSLT significa Transformaciones XSL. XSL significa Lenguaje de hojas de estilo extensible, y es un lenguaje de hojas de estilo para documentos XML.

La acción XML almacenará el documento XML convertido en la variable seleccionada. El archivo original queda intacto en el disco. Si quiere guardar el documento XML convertido, use la acción [Guardar datos en el archivo](#).



## NOTA

Comúnmente, usará la acción para simplificar los documentos XML proporcionados por la aplicación host. La definición de un filtro XML para el documento XML complejo puede tardar un tiempo o, en algunos casos, el XML puede ser demasiado complejo para administrar. Como alternativa, definirá las reglas para convertir el XML en una estructura que pueda manipular fácilmente el filtro XML, o incluso evitar por completo la necesidad de un filtro. Puede convertir el documento XML en XML nativo compatible, como XML de Oracle, y luego simplemente ejecutarlo con la acción [Ejecutar archivo de comando XML de Oracle](#).

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Origen de datos** define los datos XML a transformar.

- **Usar datos recibidos por el desencadenador:** define que el desencadenador recibió datos que usó. El mismo resultado puede archivarse habilitando la variable interna `DataFileName` y usando el contenido del archivo al que se refiere. Para obtener más información, consulte la sección Usar valores compuestos en la guía de usuario NiceLabel Automation.
- **Nombre de archivo:** define la ruta de acceso y el nombre del archivo que contiene el archivo XML para transformar. Se utiliza el contenido del archivo especificado. Origen de datos permite que el nombre del archivo se defina en forma dinámica. Seleccione o cree una variable que contenga la ruta de acceso o el nombre del archivo. La acción abrirá el archivo especificado y aplicará la transformación en el contenido del archivo, el cual debe tener el formato XML.
- **Variable:** selecciona o crea la variable que contiene la secuencia de la impresora. Se usa el contenido de la variable seleccionada y debe tener estructura XML.

El grupo **Reglas de transformación de origen de datos (XSLT)** define las reglas de transformación (documento .XSLT) que se aplicarán al documento XML.

- **Nombre de archivo:** define la ruta de acceso y el nombre del archivo que contiene las reglas de transformación (.XSLT).
- **Personalizado:** define los contenidos personalizados. Puede utilizar contenido fijo, una combinación de contenido fijo y variable, o contenido únicamente variable. Para insertar contenido variable, haga clic en el botón con flecha que se encuentra a la derecha del área de datos e inserte una variable de la lista. Para obtener más información, consulte la sección Usar valores compuestos en la guía de usuario NiceLabel Automation.



El grupo **Guardar resultado en variable** define la variable que se debe almacenar en el archivo transformado.

- **Variable:** selecciona o crea una variable que contendrá el resultado del proceso de transformación. Por ejemplo, si usa las reglas que convierten el XML complejo en XML más simple, el contenido de la variable seleccionada es un XML simple.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables **ActionLastErrorId** y **ActionLastErrorDesc**.

## 5.15.7. Grupo

Use esta acción para configurar muchas acciones dentro del mismo contenedor. Todas las acciones colocadas debajo de una acción **Agrupar** pertenecen al mismo grupo y se ejecutan juntas.

Esta acción brinda los siguientes beneficios:

- **Mejor organización y visualización del flujo de trabajo de la acción.** Puede expandir o contraer la acción Agrupar y mostrar las acciones anidadas solo cuando sea necesario. Esto ayuda a mantener más ordenada el área de configuración.
- **Definir la ejecución condicional.** Puede definir una condición en la acción Agrupar una sola vez, no individualmente para cada acción. Cuando se cumple con una condición, se ejecutan todas las acciones dentro del grupo. De este modo, se ahorra mucho tiempo de configuración y se puede reducir la cantidad de errores de configuración. La acción Agrupar es un buen método para definir la ejecución condicional SI..., ENTONCES para múltiples acciones.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.15.8. Registro de eventos

<b>EDITION AVAILABILITY:</b>	Automation Easy	Automation Pro	Automation Enterprise

Esta acción registra un evento en NiceLabel Control Center como historial y para resolver problemas.



## NOTA

Para activar la acción Registrar eventos, asegúrese de que esté habilitado el registro de trabajos de impresión NiceLabel Control Center.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Datos de evento** ofrece información acerca del evento registrado.

- **Información:** descripción básica del evento que se incluirá en el registro de eventos NiceLabel Control Center. Se permiten hasta 255 caracteres en esta área.

- **Detalles:** descripción detallada del evento que se registrará en NiceLabel Control Center. Se permiten hasta 2000 caracteres en esta área.



## SUGERENCIA

Las descripciones ingresadas en los campos **Información** y **Detalles** le permiten filtrar los eventos **Historial de todas las actividades del Control Center**. Al trabajar con el Control Center, vaya a **Historial > Todas las actividades > Definir filtro**. Lea la [Guía del usuario de Control Center para conocer más detalles](#).

## Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



## NOTA

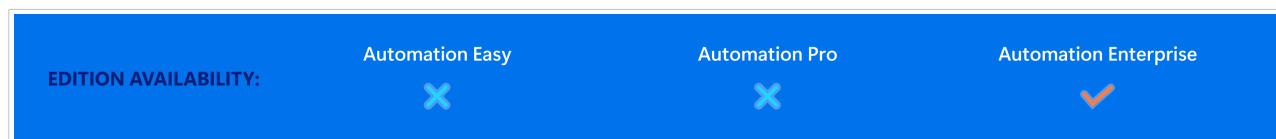
Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

## Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables **ActionLastErrorId** y **ActionLastErrorDesc**.

## 5.15.9. Generar vista previa de etiqueta



Esta acción ejecuta el proceso de impresión y proporciona una vista previa de la imagen. De manera predeterminada, la vista previa se guarda en el disco como imagen JPEG, pero se puede seleccionar otro formato de imagen. También se puede controlar el tamaño de la imagen de vista previa creada. La acción genera la vista previa de una única etiqueta.

Cuando tenga la vista previa de la etiqueta creada en un archivo, podrá enviarlo a una aplicación de terceros utilizando una de las acciones de salida, como [Enviar datos a HTTP](#), [Enviar datos a puerto de serie](#) o [Enviar datos a puerto TCP/IP](#), o bien como mensaje de respuesta de desencadenadores bidireccionales, como Desencadenador de servidor HTTP y Desencadenador de servicio Web. La aplicación de terceros puede tomar la imagen y mostrarla como vista previa de etiqueta al usuario.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Vista previa** define el archivo a previsualizar y sus detalles.

- **Nombre de archivo** especifica el nombre del archivo y la ruta. Si está preprogramados, el mismo archivo se usa cada vez. Si utiliza solo el nombre del archivo sin la ruta de acceso, se usa la carpeta en la que se guardó el archivo de configuración (.MISX). Puede usar referencias relativas para el nombre del archivo si se usa la carpeta con el archivo .MISX como carpeta raíz. **Origen de datos** habilita el nombre de archivo variable. Seleccione o cree una variable que contenga la ruta de acceso o el nombre del archivo. una vez que se ejecuta un desencadenador. Generalmente, el valor de la variable se asigna según un filtro.
- **Tipo de imagen:** especifica el tipo de imagen que se usa para guardar la vista previa de la etiqueta.
- **Vista previa del reverso de la etiqueta (etiquetas de dos lados):** permite la vista previa del reverso de la etiqueta. Esto es útil cuando se utilizan etiquetas de dos lados y se desea generar la vista previa del reverso de la etiqueta.

### Ejemplo

Por ejemplo, si su plantilla de etiquetas define las dimensiones como 4" x 3" y la impresora tiene una resolución de 200 DPI, las dimensiones de la imagen de vista previa resultante serán 800 x 600 píxeles. El

ancho equivale a 4 pulgadas multiplicadas por 200 DPI, cuyo resultado es 800 píxeles. La altura equivale a 3 pulgadas multiplicadas por 200 DPI, cuyo resultado es 600 píxeles.

El grupo **Configuración adicional** permite habilitar el uso de valores provisionales.

- **Usar valores provisionales:** reemplaza los valores faltantes del origen de datos con valores provisionales y los muestra en la vista previa de la etiqueta.



### SUGERENCIA

**Valor provisional:** define el valor variable de un marcador de posición personalizado en un objeto al diseñar etiquetas o formularios. En un objeto de etiqueta, el valor provisional se reemplaza por el valor real de la variable al momento de la impresión.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:

- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



### NOTA

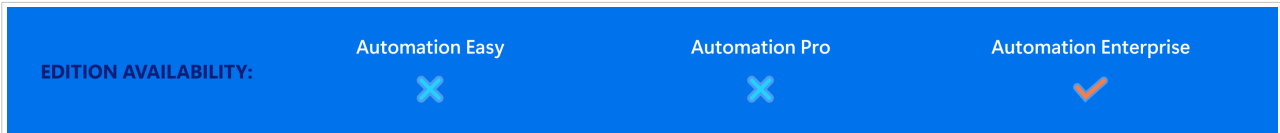
Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

## 5.15.10. Crear variante de etiqueta



Esta acción le permite crear una variante de una etiqueta existente lista para revisión. Los objetos de etiquetas en dichas variantes tienen valores de orígenes de datos bloqueados. Su contenido está definido por el valor actual del origen de datos aplicable.

El propósito de crear una variante de una etiqueta lista para revisión con orígenes de datos "bloqueados" es hacer que la etiqueta sea adecuada para el proceso un proceso de aprobación en el que los datos y las plantillas deben aprobarse juntos. En lugar de ver una etiqueta sin contenido definido para los objetos incluidos, el aprobador revisa una variante con valores definidos. Esto le permite ver y aprobar rápidamente el diseño de etiqueta final con los valores reales que se usarán para impresión.



### SUGERENCIA

El proceso de aprobación de la etiqueta se aplica a las etiquetas almacenadas en el Almacenamiento de documentos del Centro de control. Puede aplicar distintos tipos de flujo de trabajo de aprobación para las etiquetas almacenadas y las variantes de etiquetas. La selección del flujo de trabajo de aprobación depende de los requerimientos de su entorno empresarial. Consulte NiceLabel 10 Guía del usuario del Centro de control para conocer más detalles.

El grupo **Acerca de** identifica la acción seleccionada.

- **Nombre:** le permite definir un nombre de acción personalizado. Esto permite reconocer las acciones fácilmente en la lista de acciones de la solución. En forma predeterminada, el nombre de la acción se toma de su tipo.
- **Descripción:** información personalizada acerca de la acción. Ingrese una descripción para explicar el propósito y la función de la acción en una solución.
- **Tipo de acción:** información de solo lectura acerca del tipo de acción seleccionado.

El grupo **Configuración** define el archivo de etiqueta a convertir y el archivo de salida (variante de etiqueta).

- **Nombre de la etiqueta:** el nombre del archivo de la etiqueta puede convertirse a una variante lista para revisión con valores de orígenes de datos bloqueados. **El origen de datos** define en forma dinámica el **nombre de la etiqueta** usando una variable existente o recientemente creada.

- **Orígenes de datos al momento de impresión:** esta opción le permite definir orígenes de datos para los cuales el valor se proporcionará al momento de la impresión. Si un origen de datos aparece en este campo, su valor no se bloquea y puede proporcionarse al momento de impresión. Algunos ejemplos típicos son orígenes de datos para valores de producción, como número de lote, fecha de vencimiento, etc.



### SUGERENCIA

Inserte solo nombres de orígenes de datos sin corchetes, separados por comas o enumerados en una columna usando la tecla Enter.

- **Nombre de archivo de salida:** el nombre del archivo de variante de etiqueta que estará listo para revisión. **El origen de datos** define en forma dinámica el **nombre de la etiqueta** usando una variable existente o recientemente creada.

Hay distintas reglas que se aplican a la variante de etiqueta lista para revisión:

1. Los valores de orígenes de datos están bloqueados de manera predeterminada. Para que los orígenes de datos no estén bloqueados, inclúyalos en el campo **Orígenes de datos al momento de impresión** para que se mantengan activos en la etiqueta lista para revisión. Puede definir su valor al momento de impresión.
2. Las variables del contador, funciones, campos de base de datos y variables globales se convierten a variables no solicitadas.
3. Los gráficos están insertados.
4. La variante de etiqueta de destino almacenada en el NiceLabel Almacenamiento de documentos del Centro de control se protege automáticamente. El **Nombre de etiqueta** y los **Orígenes de datos al momento de impresión** originales se usan como comentario de protección.
5. Las variantes de etiquetas pueden abrirse en NiceLabel 10 Designer en estado bloqueado.
6. Los archivos de etiquetas generados con esta acción no pueden importarse.
7. Si dichas variantes de etiquetas se almacenan en la memoria de la impresora, el comando de recuperación solo puede proporcionar valores para los orígenes de datos al momento de impresión.
8. Si se usa el NiceLabel Centro de control, la vista previa de la etiqueta en Almacenamiento de documentos permite editar los orígenes de datos al momento de impresión.
9. Las variables de hora y fecha actuales no pueden configurarse como orígenes de datos al momento de impresión en la variante de etiqueta lista para revisión.

### Ejecución de acciones y tratamiento de errores

Cada acción se puede configurar como una acción condicional. Las acciones condicionales solo se ejecutan cuando las condiciones definidas permiten su ejecución. Para definir estas condiciones, haga clic en **Mostrar opciones de ejecución de acciones y tratamiento de errores**.

Las **Opciones de ejecución** son:



- **Habilitada:** Especifica si la acción está activada o desactivada. Se ejecutarán las acciones activadas únicamente. Esta funcionalidad puede usarse al probar un formulario.
- **Condición:** define una expresión de programación de una línea que debe proveer un valor booleano (**verdadero** o **falso**). Si el resultado de la expresión es **verdadero**, se ejecutará la acción. La condición ofrece una manera de evitar la ejecución de acciones cada vez.

Las opciones de **Tratamiento de errores** son:

- **Ignorar error:** especifica si un error debe ignorarse o no. Si está habilitada la opción **Ignorar error**, continúa la ejecución de acciones incluso si falla la acción actual.



#### NOTA

Las acciones anidadas que dependen de la acción actual no se ejecutan en caso de error. La ejecución de acciones continúa con la siguiente acción del mismo nivel que la acción actual. El error se registra, pero no interrumpe la ejecución de la acción.

#### Ejemplo

Al final de la impresión, tal vez quiera enviar una actualización de estado a una aplicación externa usando la acción **Solicitud HTTP**. Si la acción de impresión falla, el proceso de la acción se detiene. A fin de ejecutar el informe, incluso luego de un error de la acción de impresión, la acción **Imprimir etiqueta** debe tener la opción **Ignorar error activada**.

- **Guardar error en variable:** le permite seleccionar o crear una variable para almacenar el error. La misma causa del error también se guarda en las variables `ActionLastErrorId` y `ActionLastErrorDesc`.

# 6. Ejecutar y administrar desencadenadores

## 6.1. Implementar configuración

Una vez que configuró y probó los desencadenadores en Automation Builder, deberá implementar la configuración usando el servicio NiceLabel Automation y abrir los desencadenadores. En ese momento, los desencadenadores se activan y comienzan a monitorear eventos definidos.

Para implementar la configuración, utilice uno de los siguientes métodos.

### Implementar desde Automation Builder

1. Abra Automation Builder.
2. Cargue la configuración.
3. Vaya a la pestaña **Elementos de configuración**.
4. Haga clic en el botón **Implementar configuración** en el grupo de la banda de opciones Implementar. La configuración se cargará dentro de Automation Manager que funciona en la misma máquina.
5. Abra los desencadenadores que quiera activar.

Si esta configuración ha ya sido cargada, la implementación fuerza la recarga, manteniendo activo el estado de los desencadenadores.

### Implementar desde Automation Manager

1. Abra Automation Manager.
2. Vaya a la pestaña **Desencadenadores**.
3. Haga clic en el botón **+Agregar** y busque la configuración en el disco.
4. Abra los desencadenadores que quiera activar.

### Implementar usando la línea de comandos

Para implementar la configuración `C:\Project\Configuration.MISX` y ejecutar el desencadenador dentro del `CSVTrigger` nombrado, realice lo siguiente:

```
NiceLabelAutomationManager.exe ADD c:\Project\Configuration.MISX  
NiceLabelAutomationManager.exe START c:\Project\Configuration.MISX CSVTrigger
```

Para obtener más información, consulte el tema [Control del servicio con parámetros de línea de comandos](#).

## 6.2. Opciones de registro de eventos

NiceLabel Automation registrará eventos en varios destinos, según el escenario de implementación. Las primeras dos funciones de registro están disponibles con cada nivel de producto NiceLabel Automation.

- **Registrar a base de datos de registro:** El registro en bases de datos de registro internas siempre está habilitado. La base de datos de registro lleva el registro de todos los eventos con todos los detalles. Al ver la información registrada, puede utilizar filtros para mostrar eventos que coincidan con las reglas. Para obtener más información, consulte la sección [Usar registro de eventos](#). Los datos se almacenan en la base de datos SQLite. Es un repositorio de registro temporal, los eventos se eliminan de la base de datos semanalmente. El intervalo de mantenimiento se configura en Opciones. Los registros de eventos antiguos se eliminan de la base de datos, pero la base de datos no se compacta (vacía) de modo que sigue ocupando espacio en el disco. Para compactarla, utilice un software de administración SQLite de terceros.
- **Registrar en el registro de eventos de aplicación de Windows:** Los eventos importantes se guardan en el registro de eventos de aplicación de Windows en caso de que NiceLabel Automation no se inicie. Esto garantiza un recurso secundario para eventos registrados.
- **Registrar en Centro de control:** Se puede registrar en el Centro de control en productos de **LMS Enterprise** y **LMS Pro**. El Centro de control es una consola de administración basada en Web que registra todos los eventos desde uno o más NiceLabel Automation servidores. Los datos se almacenan en la base de datos Microsoft SQL Server. Puede buscar en los datos recopilados, y, adicionalmente, la aplicación también puede tener alertas automatizadas para ciertos eventos, administración de impresora, almacenamiento de documentos, sistema de control de revisión (control de versiones), flujo de trabajo y reimpresión de etiquetas.



### NOTA

Para obtener más información, consulte la [guía del usuario de Control Center](#).

## 6.3. Administrar desencadenadores

Automation Manager es la parte administrativa del software NiceLabel Automation. Si utiliza Automation Builder para configurar desencadenadores, utilizará Automation Manager para implementarlos y ejecutarlos en un entorno de producción. La aplicación le permite cargar desencadenadores de diferentes configuraciones, consultar el estado, iniciarlos/detenerlos y ver los detalles de configuración en el archivo de registro.

Puede personalizar la vista de las configuraciones cargadas y sus desencadenadores. Automation Manager recuerda la última vista y la aplica cuando lo ejecuta la próxima vez. Cuando habilita la vista **Por estado**, los desencadenadores de todas las configuraciones abiertas que tienen ese estado se implementan juntos. Cuando habilita la vista **Por configuración**, los desencadenadores de la configuración seleccionada se implementan juntos, independientemente de su estado. El estado está codificado por color en el icono del desencadenador para facilitar la identificación.

Los detalles del desencadenador mostrados cambian en tiempo real a medida que se detectan eventos de desencadenadores. Puede ver la información, como el nombre del desencadenador, el tipo de desencadenador, cuántos eventos se procesaron, cuántos errores se detectaron y el tiempo que pasó desde el último evento. Si desplaza el mouse sobre el número de desencadenadores ya procesados, verá la cantidad de eventos de desencadenadores que se procesarán.



### NOTA

La configuración cargada se guarda en la memoria caché. Si realiza un cambio en la configuración en Automation Builder, Automation Manager no lo aplicará automáticamente. Para aplicar el cambio, debe recargar la configuración.

### Cargar la configuración

Para cargar la configuración, haga clic en el botón **+Agregar** y busque el archivo de configuración (.MISX). Los desencadenadores de la configuración se cargarán en estado suspendido. Debe iniciar los desencadenadores para activarlos. Para obtener más información, consulte el tema [Implementar configuración](#).

Se guardan la lista de configuraciones cargadas y el estado de cada desencadenador. Si el servidor se reinicia por cualquier motivo, el Servicio NiceLabel Automation restablecerá el estado del desencadenador como estaba antes del reinicio.

### Recarga y eliminación de la configuración

Una vez que actualiza y guarda la configuración en Automation Builder, los cambios no se aplican automáticamente en Automation Manager. Para recargar la configuración, haga clic con el botón derecho en el nombre de la configuración, y seleccione **Recargar configuración**. Esta acción recarga todos los desencadenadores. Si [Almacenar archivos en caché](#) está activado, la recarga forzará la sincronización de todos los archivos utilizados por los desencadenadores.

### Iniciar / detener desencadenadores

Cuando carga desencadenadores de una configuración, se detiene el estado predeterminado. Para iniciar el desencadenador, haga clic en el botón **Iniciar** en el área del desencadenador. Para detener el desencadenador, haga clic en el botón **Detener**. Puede seleccionar más desencadenadores de la misma configuración e iniciarlos / detenerlos simultáneamente.

También puede controlar el inicio o la detención de una configuración usando la línea de comandos. Para obtener más información, consulte el tema [Control del servicio con parámetros de línea de comandos](#).

### Tratamiento de conflictos con desencadenadores

Los desencadenadores pueden quedar en estado de error debido a las siguientes situaciones. No puede iniciar esos desencadenadores hasta que no resuelva el problema.

- **Desencadenador no configurado correctamente o completamente:** En este caso, el desencadenador no está configurado, las propiedades obligatorias no están definidas o las acciones definidas para esta impresora no están configuradas. No puede iniciar dicho desencadenador.
- **La configuración del desencadenador se superpone con otro desencadenador:** Dos desencadenadores no pueden monitorear al mismo evento.

### Ejemplo

Dos desencadenadores de archivos no pueden monitorear el mismo evento. Los desencadenadores HTTP no pueden aceptar datos en el mismo puerto. Si la configuración del desencadenador se superpone con otro desencadenador, el segundo desencadenador no se ejecutará, porque el evento ya está capturado por el primer desencadenador. Para obtener más información, consulte el Panel de registro de ese desencadenador.

### Restablecer el estado de error

Cuando la ejecución de un desencadenador provoca un error, el icono del desencadenador se pone rojo, el desencadenador recibe el estado de error y los detalles de eventos se registran en la base de datos de registros. Incluso si todos los eventos siguientes se completan satisfactoriamente, el desencadenador seguirá con el estado de error hasta que confirme que comprende el error y quiere cambiar el estado. Para aceptar el error, haga clic en el icono junto al contador de errores en los detalles del desencadenador.

### Usar el panel de notificación

El panel de notificación es el área sobre la lista de desencadenadores en la pestaña Desencadenadores donde se muestran los mensajes importantes. El área de notificación mostrará los **mensajes de estado** de la aplicación, como "Modo de prueba" o "Modo de prueba expiró", o **mensajes de advertencia**, tales como "Se habilitó el rastreo".

### Ver datos registrados

Cada actividad de los desencadenadores se registra en la base de datos, incluidos los eventos de inicio/detención, la ejecución satisfactoria de acciones y los errores encontrados durante el procesamiento. Haga clic en el botón Registrar para ver los eventos registrados solo para el desencadenador seleccionado. Para obtener más información, consulte la sección [Usar registro de eventos](#).

## 6.4. Usar registro de eventos

Todas las actividades que se realizan en el software NiceLabel Automation se registran en una base de datos para historial y resolución de problemas. Cuando hace clic en el botón **Registro** en la pestaña Desencadenadores, se muestran los eventos para ese desencadenador determinado. El panel de registro mostrará la información de todos los eventos que se relacionen con el filtro definido.

Registrar los datos es útil para resolver problemas. Si el desencadenador o la acción no se pueden ejecutar, la aplicación registra una descripción del error en el archivo de registro que lo ayuda a identificar y resolver el problema.



### NOTA

El tiempo de retención de datos predeterminado es 7 días y se configura en Opciones. Para minimizar el tamaño de la base de datos de registros en sistemas ocupados, deberá reducir el período de retención.

## Filtrado de eventos

Filtros que se pueden configurar:

- **Configuración y desencadenadores:** Especifica qué eventos mostrar, eventos del desencadenador seleccionado o eventos de todos los desencadenadores de la configuración seleccionada.
- **Período registrado:** Especifica el marco de tiempo en el cual ocurren los eventos. El marco de tiempo predeterminado es **Últimos 5 minutos**.
- **Nivel de evento:** Especifica el tipo (importancia) de los eventos que quiere mostrar:
  - **Error** es el tipo de evento que quiebra la ejecución.
  - **Advertencia** es el tipo de evento en el ocurren errores, pero se configuran para que se ignoren.
  - **Información** es el tipo de evento que registra toda la información sin errores.



### NOTA

En caso de errores y advertencias, Automation también muestra la secuencia completa de acciones ejecutadas con éxito en un desencadenador.

The screenshot shows the 'Log' tab in the NiceLabel Automation Manager. The main window displays a table of events with columns for Timestamp, ID, Name, and Description. The table shows a sequence of actions: 'CSV Medium complexity' (Trigger), 'Use Data Filter', 'Action started', 'For each line' (Loop), 'Open Label', and 'Set Printer'. The 'Set Printer' action has a red error message: 'Printer name "PRA R-402" is invalid for set printer action.' An 'Options' dialog box is open, showing settings for 'Automation' and 'Performance'. The 'Log messages' section is set to 'Errors and warnings'. A green callout box points to the 'Log messages' dropdown with the text: 'Show only errors and warnings.' Another green callout box points to the 'Set Printer' error message with the text: 'Also OK actions visible. This gives context to the reported error.'

Timestamp	ID	Name	Description
5.4.2019 14:31:00.998		CSV Medium complexity	Trigger was executed - File which executes the trigger is "C:\Users\Public\Documents\NiceLabel 2019\Automation\Samples\CSV Medium\data.txt".
5.4.2019 14:31:01.008	1	Action "Use Data Filter"	Use data filter action for filter "Structural definition of DATA.TXT" is executed.
5.4.2019 14:31:01.047	1.1	For each line	Action started
5.4.2019 14:31:01.058	1.1	For each line	Loop value = 1
5.4.2019 14:31:01.058	1.1.1	Action "Open Label"	Label: C:\Users\Public\Documents\NiceLabel 2019\Automation\Samples\CSV Medium\label1.nlbl
5.4.2019 14:31:01.308	1.1.1.1	Action "Set Printer"	Printer name "PRA R-402" is invalid for set printer action.

El nivel de registro se configura en **Opciones**.

- **Filtrar por descripción:** Puede mostrar todos los eventos que contengan la cadena provista. Utilice esta opción para resolver problemas de los desencadenadores ocupados. El filtro se aplicará al campo de descripción del desencadenador.

### Borrar la base de datos de registro

Puede borrar el registro de Automation Builder. Para borrar la base de datos de registro, haga clic en el botón **Borrar registro**.

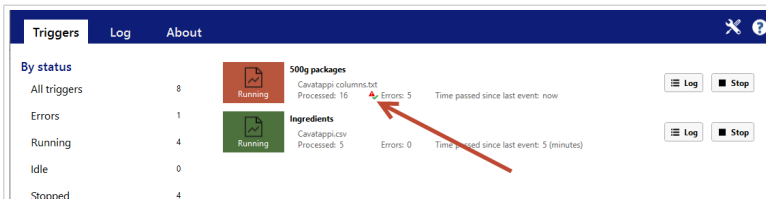


### AVISO

Use el borrado de registro con cautela, ya que es un comando irreversible. Este comando borra **TODOS** los eventos registrados de la base de datos, y se aplica a todos los desencadenadores, no solo al desencadenador actual.

### Limpiezas automáticas de la base de datos de registro

Automation le permite configurar limpiezas automáticas periódicas para entradas de registro sobre los desencadenadores ejecutados con éxito. De esta forma se asegura de que la creciente base de datos de registros reduzca el rendimiento del sistema.



Restablecer el contador no elimina los errores del registro.



## Limpiezas automáticas de bases de datos de registro

Automation le permite configurar limpiezas automáticas periódicas para entradas de registro sobre los disparadores ejecutados con éxito. Así es como se asegura de que la creciente base de datos de registro reduzca el rendimiento del sistema.

Para programar las limpiezas automáticas de la base de datos de registro:

1. Abrir documento `producto.config` en editor de texto.

El archivo se encuentra aquí:

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config
```

2. Crear una copia de seguridad de la `producto.config` expediente.
3. Automation utiliza dos parámetros para limpiar los disparadores. Agregue estos dos parámetros a su archivo `product.config`.
  - `/IntegrationService/LogSuccessfulTriggerPurgeInterval`. Este parámetro define la duración del intervalo de tiempo entre dos limpiezas consecutivas. Escriba la duración del intervalo en minutos.
  - `/IntegrationService/LogSuccessfulTriggerPurgeRemovalAge`. Este parámetro verifica la antigüedad de los mensajes sobre las acciones ejecutadas con éxito.

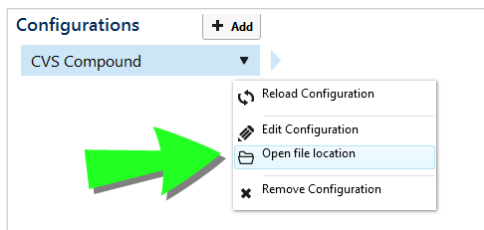
```
<configuration>
  <IntegrationService>
    <LogSuccessfulTriggerPurgeInterval>1</
LogSuccessfulTriggerPurgeInterval>

    <LogSuccessfulTriggerPurgeRemovalAge>1<LogSuccessfulTriggerPurgeRemovalAge>
  </IntegrationService>
</configuration>
```

## 6.5. Si su configuración no se carga...

Al implementarla, la configuración de Automation se ejecuta como un proceso de Windows en segundo plano. El Automation Manager que utiliza para administrar y monitorear su configuración es solo una interfaz que representa los servicios reales de Automation. En ciertos casos, la configuración que desarrolló, probó e implementó no se carga. Hay varias razones posibles. Siga las soluciones propuestas para configurar y ejecutar Automation:

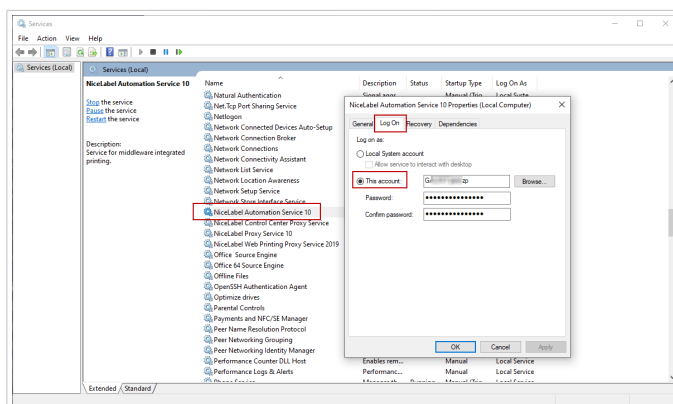
1. El archivo de configuración se eliminó, tiene otro nombre o se cambió de ubicación. Puede verificar en qué archivo están implementados sus puntos de configuración en Automation Manager:



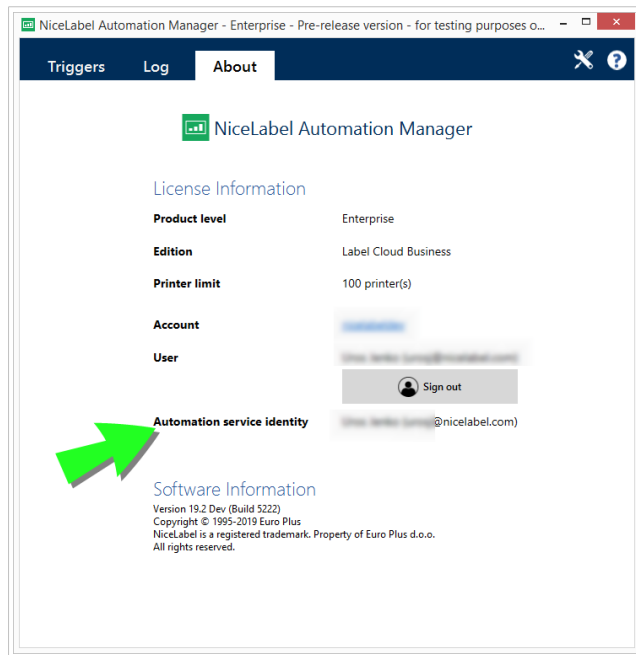
Asegúrese de que el archivo de configuración .mixs esté disponible en la ubicación especificada y que aún tenga el mismo nombre que en Automation Manager. Si movió o cambió el nombre de su archivo de configuración, ábralo en Automation Builder y vuelva a implementar la configuración.

2. El archivo de configuración se encuentra en una ubicación de la red a la que no se puede acceder debido a un problema de conexión de red. Verifique la conexión a la red de su computadora o servidor que almacena la configuración.
3. **El servicio de automatización no cuenta con el permiso para acceder al archivo de configuración.** Verifique los permisos para la cuenta de usuario utilizada por el servicio de automatización. Este error indica problemas con el servicio de Automation que se ejecuta en segundo plano. Las posibles soluciones son:

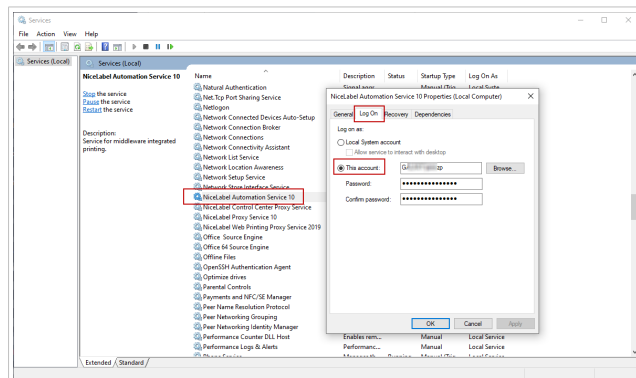
- Si el archivo de configuración está almacenado en un disco local o recurso compartido de red, verifique que esté ejecutando su configuración con sus credenciales de usuario o dominio locales. Ejecutar su configuración en Cuenta local del sistema puede restringir el acceso a las carpetas e impresoras de red compartidas. Abra Servicios y vea las propiedades de NiceLabel Automation Service 10.



- Si su archivo de configuración está almacenado en el almacenamiento de documentos de su Control Center, estos son los posibles escenarios:
  - Su Control Center usa la autenticación de la aplicación. La razón por la cual su configuración no se carga es la incorrecta **identidad del servicio de automatización**. La identidad del servicio de automatización debe coincidir con la identidad del usuario definida en su Control Center. Puede encontrar la identidad del servicio de automatización de su configuración en **Automation Manager > pestaña Acerca de**.



- Su Control Center usa la autenticación de Windows. Su configuración no se carga porque intenta ejecutarla como un usuario con privilegios insuficientes en su Control Center. Verifique con qué usuario está ejecutando el servicio Automation. Abra **Servicios** y vea las propiedades de **NiceLabel Automation Service 10**.






## NOTA

En ambos casos, consulte la Guía del usuario de NiceLabel Control Center para conocer más detalles de los métodos de autenticación y los privilegios de usuario disponibles.

# 7. Opciones de rendimiento y comentarios

## 7.1. Procesamiento paralelo

<b>EDITION AVAILABILITY:</b>	Automation Easy	Automation Pro	Automation Enterprise
			

NiceLabel Automation admite procesamiento paralelo, tanto para procesamiento de entrada como de salida. Esto garantiza máxima eficiencia en cualquier sistema con software instalado. NiceLabel Automation ejecuta múltiples tareas simultáneamente, conservando a la vez el orden en el que se han agregado los desencadenadores. La productividad del procesamiento de trabajos de etiquetas depende enormemente del hardware en uso.

### Procesamiento paralelo de entrada

Puede ejecutar múltiples desencadenadores en la misma máquina. Todos responden simultáneamente a los cambios en los eventos monitoreados. Cada desencadenador recuerda los datos de sus eventos sin procesar en la lista de cola. Esta lista almacena los datos entrantes en caso de que ningún proceso de impresión esté disponible en ese momento. En cuanto un proceso de impresión esté disponible, se toma el primer trabajo de la lista usando el principio FIFO (primero en entrar, primero en salir). Esto asegura el orden correcto del procesamiento de datos de entrada. No obstante, no garantiza el principio FIFO para imprimir. Consulte la siguiente sección a continuación.



### NOTA

El procesamiento paralelo se refiere a más que simplemente ejecutar múltiples desencadenadores a la vez. Además, cada desencadenador puede permitir conexiones simultáneas. Los desencadenadores de servicios Web, TCP/IP y HTTP aceptan todas las conexiones simultáneas de varios clientes. Asimismo, el desencadenador de archivos puede configurarse para que controle un grupo de archivos en una carpeta. Esto puede configurarse con máscara de archivos.

### Procesamiento paralelo de salida

Generalmente, el resultado del desencadenador es un proceso de impresión de etiquetas. Para este proceso, está usando datos recibidos por el desencadenador para imprimirlos en etiquetas. El servicio NiceLabel Automation ejecuta procesos de impresión (llamados "motores de impresión") en paralelo en segundo plano. Los procesadores modernos tienen dos o más unidades de procesamiento centrales independientes llamadas "núcleos". Los núcleos múltiples pueden ejecutar múltiples instrucciones al mismo tiempo, lo que incrementa la velocidad general del procesamiento. En caso de NiceLabel Automation, los núcleos múltiples incrementan el procesamiento del trabajo de impresión y, por último, el rendimiento de la impresión de etiquetas.

De manera predeterminada, cada instancia de NiceLabel Automation ejecuta cada proceso de impresión como su proceso separado en cada núcleo disponible. Cuánto más eficaz es la CPU, mayor es la capacidad de proceso. Esto maximiza el uso de la capacidad de la CPU disponible. El software se instala con valores predeterminados razonables que definen que cada núcleo disponible acepta único su proceso para procesamiento de impresión. En circunstancias normales no es necesario hacer cambios en los valores predeterminados. Si la configuración requiere un cambio, consulte la sección [Cambiar configuración predeterminada de impresión multiproceso](#).

Cuando haya muchos procesos de impresión disponibles, los datos del primer evento pueden imprimirse con un proceso de impresión, mientras que los datos del segundo evento se imprimirán con un proceso de impresión diferente simultáneamente si hay disponible un segundo proceso en ese momento. Si el segundo evento no brindó muchos datos, el proceso de impresión puede brindar los datos para la impresora más rápido que el primer proceso de impresión, lo cual cambia el orden. En ese caso, los datos del segundo evento pueden imprimirse antes que los del primer evento. Para garantizar el principio FIFO también para impresión consulte la sección [Modo de impresión sincronizada](#).

## 7.2. Almacenar archivos en caché

Para mejorar el tiempo hasta la primera etiqueta y el rendimiento general, NiceLabel Automation admite el almacenamiento en caché de archivos. Cuando cargue las etiquetas, las imágenes y los datos de la base de datos de recursos compartidos de red, es posible que haya retrasos al imprimir las etiquetas. NiceLabel Automation debe tomar todos los archivos necesarios antes de que comience el proceso de impresión.

Existen dos niveles de almacenamiento en caché que se complementan el uno al otro.

- **Memoria caché:** La memoria caché almacena los archivos ya utilizados. Las etiquetas que se usaron al menos una vez se cargan en la memoria caché. Cuando el desencadenador solicita la impresión de la misma etiqueta, la etiqueta está disponible de inmediato para el proceso de impresión. La memoria caché se activa en forma predeterminada. Su contenido se borra cuando quita o vuelve a cargar una configuración. Se comprueba si hubo cambios en el archivo de etiquetas para cada acción Abrir etiqueta. Si hay una etiqueta más nueva, se carga automáticamente, y se reemplaza la versión anterior en la memoria caché.



## NOTA

Cuando una etiqueta no se utiliza por 8 horas, se descargará de la memoria caché.

- **Caché permanente:** El caché permanente almacena datos en el disco; su rol es ofrecer almacenamiento de archivos a mediano plazo. El almacenamiento en caché se administra por objeto de archivo. Cuando se solicita un archivo del recurso compartido de red, el servicio primero verifica si el archivo ya está presente en caché y lo usa. Si el archivo no está en caché, se toma del recurso compartido de red y se almacena en caché para uso futuro. El servicio de caché actualiza continuamente el contenido en caché con las versiones nuevas de los archivos. Puede configurar los intervalos de tiempo para la verificación de la versión en el menú Opciones.

## Prolongar el período de tiempo para la descarga de etiquetas

Una vez que la etiqueta se usa por primera vez, se carga en la memoria caché. La etiqueta queda disponible para impresión instantánea la próxima vez que se requiera. El proceso de mantenimiento de la memoria caché eliminará todas las etiquetas que no se hayan utilizado por 8 horas o más.

Para prolongar el intervalo de tiempo en el que se descargarán las etiquetas de la memoria caché, realice lo siguiente:

1. Navegue a la carpeta NiceLabel Automation System.  
%PROGRAMDATA%\NiceLabel\NiceLabel 10
2. Realice una copia de seguridad del archivo `product.config`.
3. Abra `product.config` en el editor de texto. El archivo tiene una estructura XML.
4. Agregue el elemento `Common/FileUpdater/PurgeAge`.
5. Este parámetro define la cantidad de segundos que definen el marco de tiempo para conservar las etiquetas en la memoria caché. NiceLabel Automation mantiene un registro del tiempo que se utilizó cada etiqueta para imprimir por última vez. Cuando ese marco de tiempo alcanza el umbral definido, las etiquetas se descargan de la memoria.



### NOTA

Valor predeterminado: 28800 (8 horas). El valor máximo es 2147483647.

El archivo `product.config` debe tener el siguiente contenido:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <Common>
    <FileUpdater>
      <PurgeAge>28800</PurgeAge>
    </FileUpdater>
  </Common>
  ...
</configuration>
```

6. Después de guardar el archivo, el Servicio NiceLabel Automation aplica automáticamente la configuración.

## Habilitar caché permanente

EDITION AVAILABILITY:	Automation Easy ✗	Automation Pro ✗	Automation Enterprise ✓
-----------------------	----------------------	---------------------	----------------------------

Para habilitar y configurar la memoria caché permanente, abra Opción, seleccione NiceLabel Automation y habilite los **archivos remotos en caché**.

- **Actualizar archivos en caché:** Define el intervalo de tiempo en minutos en el que los archivos en caché se sincronizan con los archivos de la carpeta original. Es el intervalo de tiempo que da al sistema para usar la versión anterior del archivo.
- **Eliminar archivos en caché con más de:** Define el intervalo de tiempo en días para eliminar todos los archivos en caché que no se han accedido durante una duración determinada de tiempo.

NiceLabel Automation utiliza la siguiente carpeta local para copiar en caché los archivos remotos:

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10\FileCache
```



### NOTA

El almacenamiento de archivos en caché admite formatos de archivos de etiquetas e imágenes. Después de habilitar el almacenamiento de archivos en caché, reinicie el servicio Automation para que los cambios surtan efecto.

## Forzar la recarga del contenido en caché

NiceLabel Automation actualiza automáticamente el contenido en caché en el intervalo de tiempo definido. El valor predeterminado es 5 minutos.

Para forzar manualmente la recarga de caché, realice lo siguiente:

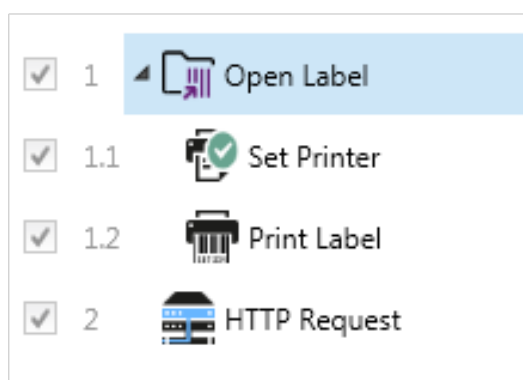
1. Abra Automation Manager.
2. Busque la configuración que contiene el desencadenador, para el cual quiera forzar la recarga de etiquetas.
3. Haga clic con el botón derecho en la configuración.
4. Seleccione **Recargar configuración**.



## 7.3. Tratamiento de errores

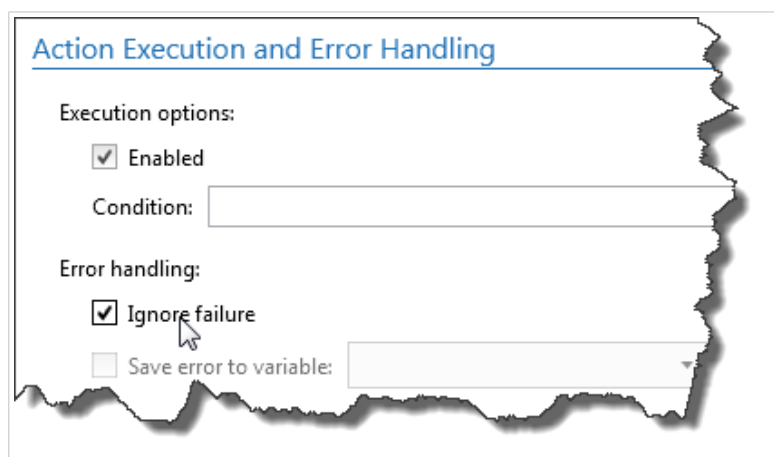
Cuando ocurre un error durante la ejecución de alguna acción, NiceLabel Automation detiene la ejecución de todas las acciones del desencadenador. Si hay acciones definidas después de la acción actual que informa un error, estas acciones no se ejecutan.

Por ejemplo, las acciones se definen como se muestra en la captura de pantalla. Si falla la acción **Configurar impresora** porque se brindó un nombre inválido o una impresora inaccesible, no se ejecutarán las acciones **Imprimir etiqueta** y **Solicitud HTTP**. El procesamiento de acciones se detendrá en **Configurar impresora**, Automation Manager mostrará el desencadenador con el estado de error y los comentarios del estado del desencadenador (si está activado) serán "impresora especificada incorrecta / impresora no accesible".



No obstante, en este caso en particular, no es recomendable usar comentarios sincronizados (enviados automáticamente cuando se activan en el desencadenador que admite comentarios sincronizados). Los comentarios de estado deben brindarse de manera asincrónica con la acción **Solicitud HTTP** después de crear el trabajo de impresión (o no). Después de que el proceso de impresión se completa, actualice una aplicación con su estado. Para esto, envíe a la aplicación un mensaje en formato HTTP.

En este caso, la acción **Solicitud HTTP** debe ejecutarse independientemente del éxito de todas las acciones de la lista arriba de esta. Debe activar la opción **Ignorar error** para todas las acciones que están arriba de la acción **Solicitud HTTP**. La opción está disponible en las opciones **Ejecución y tratamiento de errores** de una acción.



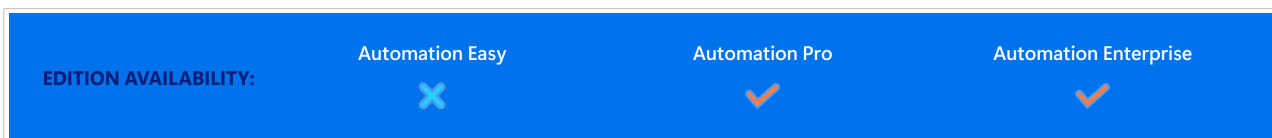
Si falla una acción específica, NiceLabel Automation comenzará a ejecutar la siguiente acción en el nivel de jerarquía superior.

### Ejemplo

Si la acción **Establecer impresora** en el nivel 1.1 falla, la ejecución no continúa con la acción **Imprimir etiqueta** en el nivel 1.2 debido a que probablemente también falle. Continúa con la acción **Solicitud HTTP** en el nivel 2, debido a que es la siguiente acción en la jerarquía de nivel superior.

Se puede implementar la misma lógica para repetir acciones, como **Usar filtro de datos**, **Bucle** y **Para cada registro**. Con estas acciones, itera por todos los miembros de la lista. Si el procesamiento de un miembro falla por algún motivo, en forma predeterminada NiceLabel Automation detiene el procesamiento del resto de los miembros y arroja un error. Si activa la opción **Ignorar error**, se detendrá el procesamiento del miembro fallido, pero NiceLabel Automation continuará con el siguiente miembro. Al final, el error se informa de todos modos.

## 7.4. Modo de impresión sincronizada



### 7.4.1. Modo de impresión asincrónica

El modo de operación predeterminado NiceLabel Automation es el modo asincrónico. El modo asincrónico es una forma de impresión, durante la cual un desencadenador envía datos para impresión y cierra la conexión con el subsistema de impresión. El desencadenador no espera el resultado del proceso de impresión y no recibe ningún comentario. Inmediatamente después del envío de datos, el desencadenador está listo para aceptar una nueva secuencia de datos entrantes.

El modo asincrónico mejora el rendimiento del desencadenador y aumenta el número de desencadenadores que pueden procesarse en un marco de tiempo. Cada proceso de impresión tiene un búfer adelante - aquí es donde el desencadenador alimenta las solicitudes de impresión. El búfer ajusta los picos del desencadenador y se asegura de no perder datos.

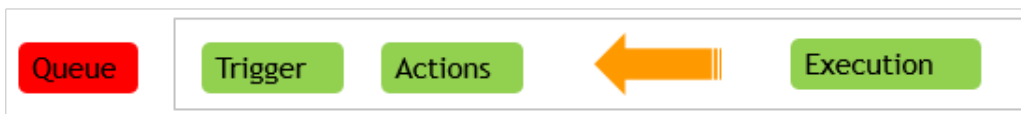
Si el error ocurre durante el procesamiento, seguirá registrado en Automation Manager (y NiceLabel Control Center, si lo utiliza), pero el desencadenador no lo reconoce. Al usar Automation en el modo de impresión asincrónica, no puede definir las acciones condicionales que ejecutaría si la ejecución del desencadenador presenta un error.



## 7.4.2. Modo de impresión sincronizada

En comparación con el modo asincrónico, el modo sincronizada no interrumpe la conexión cuando comienza el proceso de impresión. En este modo, el desencadenador envía los datos para imprimir y mantiene la conexión al subsistema de impresión establecido siempre que esté ocupado ejecutando acciones. Cuando finaliza el proceso de impresión (con o sin errores), el desencadenador recibe comentarios sobre el estado.

Puede usar esta información dentro de las acciones definidas en el mismo desencadenador y tomar la decisión de ejecutar otras acciones en caso de que haya ocurrido un error. También puede enviar el estado del trabajo de impresión a la aplicación que emite los datos. Para obtener más información, consulte la sección [Comentarios sobre el estado del trabajo de impresión](#).



### Ejemplo

Puede informar el estado de impresión a la aplicación ERP que proporcionó los datos.

## Habilitar el modo de impresión sincronizada

El modo sincronizado se define por desencadenador. Para habilitar el modo sincronizado en un desencadenador, haga lo siguiente:

1. Abra las propiedades del desencadenador.
2. Vaya a la pestaña **Configuración**.
3. Seleccione **Otro**.
4. En la sección **Comentarios del motor de impresión**, habilite la opción **Impresión supervisada**.



### NOTA

Cuando el desencadenador se ejecuta en el modo de impresión sincronizada, se comunica con un solo proceso de impresión. Al habilitar el modo de impresión sincronizada se garantiza el método FIFO para la manipulación de eventos en la dirección de salida (impresión). De manera predeterminada, el procesamiento de múltiples núcleos no puede asegurar el orden de impresión.

De manera predeterminada, el procesamiento de múltiples núcleos no puede asegurar el orden de impresión. Al cambiar la configuración predeterminada de la impresión multiproceso y limitar las solicitudes concurrentes, se puede garantizar el método FIFO:

1. Abra el archivo `product.config` en el editor de texto.  
El archivo se encuentra aquí:

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config
```

2. Establezca los valores de los elementos **ProcesosDelImpresiónSimultáneosMáx** y **ProcesosDelImpresiónDeSesionesSimultáneosMáx** a 1.

```
<configuration>
  <IntegrationService>
    <MaxConcurrentPrintProcesses>1</
MaxConcurrentPrintProcesses>
    <MaxConcurrentSessionPrintProcesses>1</
MaxConcurrentSessionPrintProcesses>
  </IntegrationService>
</configuration>
```

3. Guarde el archivo. Nicelabel Automation actualizará automáticamente el servicio con el nuevo número de subprocesos de impresión.
4. Abra el Builder y vaya a las propiedades del desencadenador.
5. Vaya a la pestaña **Ejecución**.

6. En la sección **Solicitudes concurrentes**, establezca el **Número de ejecuciones de acciones concurrentes** en 1.

## 7.5. Comentarios sobre el estado del trabajo de impresión

La aplicación que brinda los datos para la impresión de etiquetas en NiceLabel Automation puede necesitar información sobre el estado del trabajo de impresión. Los comentarios pueden ser tan simples como "Correcto" en caso de que la generación del trabajo de impresión sea satisfactoria, o se puede incluir una descripción del error detallada en caso de que haya problemas. Por motivos de rendimiento, NiceLabel Automation deshabilita los comentarios predeterminados. De esta manera, se garantiza una impresión de alto rendimiento, ya que el desencadenador no está relacionado con la ejecución del proceso de impresión. Los errores se registrarán en la base de datos de registros, pero el desencadenador no los controla.

También puede usar este método para enviar comentarios acerca de otros datos que recopila el desencadenador. Estos pueden ser el estado de las impresoras de red, el número de trabajos en el administrador de trabajos de impresión de la impresora, la lista de etiquetas de una carpeta, la lista de variables del archivo de etiquetas especificado y mucho más.



### NOTA

Para habilitar el soporte de comentarios del motor de impresión, debe habilitar el modo de impresión sincronizada. Para obtener más información, consulte el tema [Modo de impresión sincronizada](#).

Habilite los comentarios del estado del trabajo de impresión usando uno de los dos métodos disponibles.

## **El desencadenador proporciona comentarios sobre el estado del trabajo de impresión (comentarios sincronizados)**

Algunos desencadenadores tienen incorporada la función de comentarios. Si el modo de impresión sincronizada está habilitado, el desencadenador reconoce internamente el estado del trabajo. El cliente puede enviar los datos al desencadenador, mantener la conexión abierta y esperar los comentarios. Para usar este método de comentarios, seleccione y use un tipo de desencadenador que permita proporcionar comentarios.

Cuando ocurre un error en una de las acciones, la variable interna `ActionLastErrorDesc` incluirá el mensaje de error detallado. Puede enviar el valor como está o personalizarlo.

Para obtener más información, consulte los detalles acerca de los tipos de desencadenador respectivo.

- **Desencadenador de servicio Web:** El desencadenador admite comentarios por diseño. El documento WSDL (Lenguaje de descripción de servicios Web) describe los detalles sobre la interfaz del servicio Web y explica cómo habilitar los comentarios. Puede usar la respuesta predeterminada que devuelve la descripción del error en caso de que falle la acción de impresión. También puede personalizar la respuesta y enviar el contenido de cualquier variable. La propia variable puede tener datos, incluida la vista previa de las etiquetas o el trabajo de impresión de etiquetas (datos binarios).
- **Desencadenador de servidor HTTP:** El desencadenador admite comentarios por diseño. NiceLabel Automation usará los códigos de respuesta HTTP estándares para indicar el estado del trabajo de impresión. También puede personalizar la respuesta HTTP y enviar el contenido de cualquier variable. La propia variable puede tener datos, incluida la vista previa de las etiquetas o el trabajo de impresión de etiquetas (datos binarios).
- **Desencadenador de servidor TCP/IP:** Este desencadenador admite comentarios, pero no automáticamente. Para hacer que entregue comentarios, debe configurar el cliente que brinda los datos de modo que no interrumpa la conexión una vez que se envíen los datos. Cuando finaliza el proceso de impresión, la siguiente acción de la lista (Enviar datos al puerto TCP/IP) podría tener habilitada la configuración Responder al remitente. Proporcionar comentarios sobre la conexión todavía abierta establecida.

## La acción proporciona comentarios sobre el estado del trabajo de impresión (comentarios asincrónicos)

En el caso de desencadenadores que no admiten comentarios en forma nativa, o si quiere enviar mensajes con comentarios durante el proceso del desencadenador, puede definir una acción que envíe comentarios a algún destino seleccionado. En este caso, la aplicación que brinda los datos puede cerrar la conexión tan pronto como se envíen los datos del desencadenador.

### Ejemplo

Utilizó el desencadenador TCP/IP para captar datos. El cliente eliminó la conexión inmediatamente después de enviar los datos, de modo que no se puede responder dentro de la misma conexión. En dichos casos, puede usar algún otro canal para enviar comentarios. Puede configurar cualquiera de las acciones de conexión de salida, como [Ejecutar instrucción SQL](#), [Abrir documento / programa](#), [Solicitud HTTP](#), [Enviar datos al puerto TCP/IP](#) y otras. Colocará dicha acción debajo de la acción [Imprimir etiqueta](#).

Si quiere enviar comentarios únicamente de estados específicos, como "ocurrió un error", puede usar los siguientes métodos.

- **Usar condición sobre acción:** El estado del trabajo de impresión está expuesto en dos [variables internas](#) (`ActionLastErrorID` y `ActionLastErrorDesc`). La primera tendrá el ID de error o el valor 0 si no hay errores informados. La segunda tiene el mensaje de error detallado. Puede usar los valores de esas variables en ciertas condiciones en las acciones que quiera ejecutar en caso de errores. Por ejemplo, puede usar la acción **Solicitud HTTP** después de imprimir. La acción podría enviar comentarios si se produjera un error. Para habilitar dichos comentarios, realice lo siguiente:

1. Abra las propiedades del desencadenador.
2. En el grupo de la banda de opciones de **Variable**, haga clic en el botón **Variables internas** y habilite la variable `ActionLastErrorID`.
3. Vaya a la pestaña **Acciones**.
4. Agregue la acción **Enviar datos a HTTP**.
5. Dentro de las propiedades de la acción, expanda las **opciones de Mostrar ejecución y tratamiento de errores**.
6. En **Condición**, ingrese lo siguiente. La acción con esta condición se ejecutará únicamente cuando ocurra un error y la variable `ActionLastErrorID` tiene la ID del error (cualquier valor superior a 0). En forma predeterminada, la condición se ejecuta usando la sintaxis de VBScript.

```
ActionLastErrorID > 0
```

7. También debe habilitar la opción **Ignorar falla** en cada acción en la que espera la falla. Esto indica a Automatización que no deje de ejecutar por completo las acciones, sino que continúe con la siguiente acción del mismo nivel jerárquico.



## NOTA

Para obtener más información, consulte el tema [Tratamiento de errores](#).

- **Usar la acción Intentar:** La acción Intentar trata de eliminar la necesidad de condiciones de codificación. La acción proporciona dos marcadores de posición. El marcador de posición **Hacer** tendrá las acciones que quiera ejecutar. Si ocurre algún error cuando las ejecute, se interrumpirá la ejecución y se ejecutarán las acciones del marcador de posición **Al ocurrir un error**. Usará las acciones de conexión de salida en este marcador de posición para brindar un comentario sobre el estado del trabajo de impresión. Para obtener más información, consulte el tema [Intentar](#).

## 7.6. Exclusión de impresoras de la impresión automática

En ciertos casos, su entorno de impresión requiere que excluya algunas de sus impresoras del proceso de impresión automático. Las posibles razones por las que debe excluir las impresoras de la impresión automática son la política de impresión de su empresa o las limitaciones de su licencia.

Por defecto, Automation evita la impresión automática utilizando impresoras de archivos, como el controlador de fax compartido de Microsoft, Microsoft Print To PDF, Microsoft XPS Document Writer y productos similares. Estas impresoras de archivos requieren que los usuarios seleccionen manualmente la ubicación para sus «impresiones». La solicitud de intervención manual del usuario provoca que el motor de impresión se detenga e informe un error.

Para evitar que Automation use impresoras específicas en las configuraciones en ejecución:



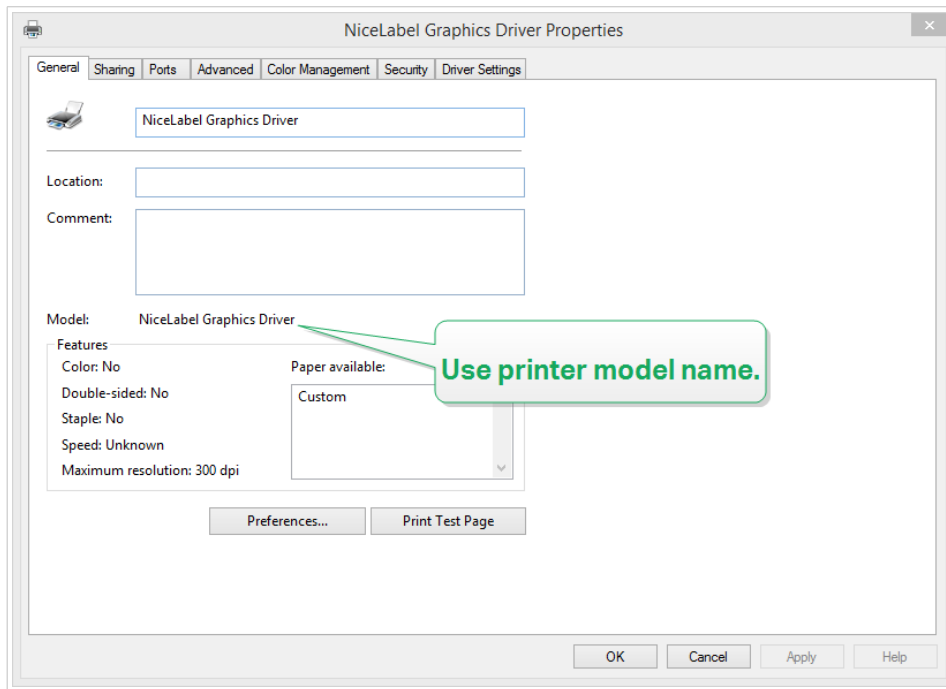
## NOTA

Al especificar las impresoras excluidas de la impresión automática en el archivo `product.config`, también debe enumerar explícitamente sus impresoras de archivos.

1. Abra el archivo `product.config` en el editor de texto.  
El archivo se encuentra aquí:  

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config
```
2. Cree una copia de seguridad del archivo `product.config`.
3. Automation utiliza dos parámetros para verificar qué modelo de impresora y puerto de impresora deben excluirse de la impresión automática. Agregue estos parámetros a su archivo `product.config`.
  - `/IntegrationService/DisabledPrinterDrivers` y escriba los modelos de impresora que desea excluir de la impresión automática.
  - `/IntegrationService/DisabledPrinterPorts` y escriba los puertos que desea excluir de la impresión automática.





```

<configuration>
  <IntegrationService>
    <DisabledPrinterDrivers>NiceLabel Graphics
Driver,NicePrinter1200dpi</DisabledPrinterDrivers>
    <DisabledPrinterPorts>LPT1,LPT2</DisabledPrinterPorts>
  </IntegrationService>
</configuration>

```

- Una vez que define los modelos de impresora y los puertos de impresora en el archivo `product.config`, aún puede ejecutar sus configuraciones de Automation, pero sus configuraciones actualizadas impiden la impresión en las impresoras mencionadas. Automation informa un error si dichas impresoras forman parte de las configuraciones en ejecución.



#### NOTA

La impresión automática no se detiene si las impresoras excluidas se utilizan en la acción [Redireccionar impresión a archivo](#).

## 7.7. Usar modo de impresión almacenar/recuperar

El modo de impresión almacenar y recuperar optimiza el proceso de impresión. Mejora la respuesta de la impresora reduciendo la cantidad de datos que necesitan enviarse durante las tareas de impresión repetitivas.

Con el modo almacenar y recuperar activado, NiceLabel Automation no necesita reenviar todos los datos de la etiqueta completos para cada impresión. En su lugar, las etiquetas (plantillas) se almacenan en

la memoria de la impresora. Los objetos fijos se almacenan como tales, mientras que los marcadores de posición se definen para los objetos variables. NiceLabel Automation solo envía datos para objetos de etiquetas variables y los comandos de recuperación. La impresora aplica los datos recibidos en los marcadores de posición de la etiqueta almacenada e imprime la etiqueta, recuperándola de la memoria. Comúnmente, se envían algunos bytes de datos a la impresora, en comparación con los kilobytes que se envían durante la normal impresión.

La acción consta de dos procesos:

- **Almacenar etiqueta:** Durante este proceso, la aplicación crea una descripción de la plantilla de la etiqueta con formato en el idioma de comando de la impresora seleccionada. Al finalizar, la aplicación envía el archivo de comando creado a la memoria de la impresora y se almacena. Puede almacenar una etiqueta desde el diseñador o desde NiceLabel Automation usando la acción [Almacenar etiqueta en impresora](#).



#### NOTA

La etiqueta debe tener definido el modo almacenar y recuperar en sus propiedades antes de poder almacenarla en la impresora.

- **Recuperar (imprimir) etiqueta:** Una etiqueta almacenada en la memoria de la impresora se imprime inmediatamente. Usando el proceso de recuperación, NiceLabel Automation crea otro archivo de comando para indicar a la impresora qué etiqueta de la memoria debe imprimir. La cantidad real de datos enviados a la impresora depende de la situación actual. Para las etiquetas fijas sin contenido variable, el archivo de comando de recuperación solo tiene el comando de recuperación de etiquetas. Para las etiquetas variables que tienen campos variables, el archivo de comando incluye los valores para esas variables y el comando de recuperación de etiquetas. Para recuperar una etiqueta desde NiceLabel Automation utilice una de las acciones de impresión comunes. Una vez ejecutada, la acción analiza la etiqueta y habilita el modo de impresión apropiado: impresión normal o impresión de recuperación, según lo definido en la etiqueta.



#### AVISO

Antes de activar este modo, asegúrese de que esté seleccionado el controlador correcto para la impresora de etiquetas. No todas las impresoras de etiquetas tienen capacidad de usar el modo de impresión almacenar y recuperar. El controlador de la impresora para la cual se creó la etiqueta en el diseñador de etiquetas también debe estar instalado en la máquina donde se ejecuta NiceLabel Automation.

## 7.8. Clúster (conmutación por error) de alta disponibilidad

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
	✗	✗	✓

NiceLabel Automation es compatible con el clúster (conmutación por error) de alta disponibilidad de Microsoft. El clúster de conmutación por error es un grupo de computadoras independientes que trabajan juntas a fin de incrementar la disponibilidad de impresión de etiquetas a través de NiceLabel Automation. Los servidores agrupados (llamados nodos) están conectados por cables físicos y por software. Si uno o más de los nodos del clúster fallan, otros nodos comienzan a brindar servicio (proceso conocido como conmutación por error). Además, los roles agrupados se monitorean proactivamente para verificar que funcionen bien. Si no funcionan, se reinician o se mueven a otro nodo. Los clientes que brindan datos se conectarán a la dirección IP perteneciente al clúster, no a las direcciones IP del nodo individual.

Para habilitar NiceLabel Automation para que funcione en un nodo de alta disponibilidad, haga lo siguiente:

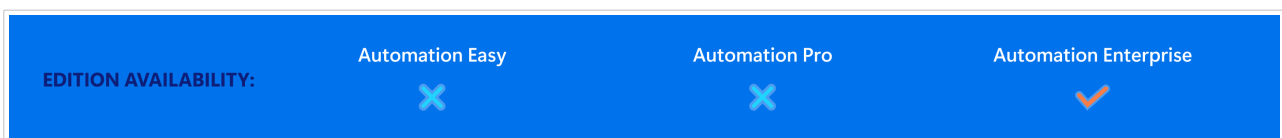
- Configure la función Clústeres de conmutación por error de Microsoft en los servidores de Windows.
- Instale NiceLabel Automation en cada nodo.
- Habilite el soporte de clústeres de conmutación por error en las propiedades NiceLabel Automation de cada nodo.

Realice lo siguiente:

1. Abra **Archivo > Opciones > Automatización**.
  2. En el grupo **Soporte de clústeres**, habilite **Soporte de clústeres de conmutación por error**.
  3. Busque la carpeta, ubicada fuera de ambos nodos, pero accesible con privilegios de acceso total al software NiceLabel Automation. Los archivos importantes del sistema que necesitan ambos nodos se copiarán a esta carpeta.
- Configure el clúster para iniciar NiceLabel Automation en el segundo nodo en caso de que esté apagado el nodo maestro.

Consulte la descripción detallada sobre la configuración del clúster de conmutación por error de automatización en nuestro [artículo de base de conocimientos](#).

## 7.9. Clúster de equilibrio de carga



NiceLabel Automation es compatible con el clúster de equilibrio de carga de Microsoft. El clúster de equilibrio de carga es un grupo de computadoras independientes que trabajan juntas a fin de incrementar la disponibilidad y la escalabilidad de impresión de etiquetas a través de NiceLabel Automation. Los servidores agrupados (llamados nodos) están conectados por cables físicos y por software. Las solicitudes entrantes de impresión de etiquetas se distribuyen entre los nodos en un clúster. Los clientes que brindan datos se conectarán a la dirección IP perteneciente al clúster, no a las direcciones IP del nodo individual.



## NOTA

Puede usar desencadenadores basados en TCP/IP con clúster de equilibrio de carga. Estos desencadenadores son [Desencadenador de servidor TCP/IP](#), [Desencadenador de servidor HTTP](#), [Desencadenador de servicio Web](#) y [Desencadenador en la nube](#).

A fin de habilitar NiceLabel Automation para el equilibrio de carga, debe hacer lo siguiente:

- Configure la función Clústeres de equilibrio de carga de Microsoft en los servidores de Windows.
- Instale NiceLabel Automation en cada nodo.
- Cargue los mismos archivos de configuración en Automation Manager en cada nodo.

# 8. Comprender las estructuras de datos

Esta sección demuestra la estructura de datos que se utiliza con frecuencia en escenarios de automatización. Al trabajar con distintos archivos de datos, debemos analizar su estructura, extraer los valores relevantes de los campos de interés e imprimirlos en etiquetas. Cada uno de los casos detallados a continuación se usa en ejemplos de configuraciones que se incluyen en el Automation paquete de instalación.

- [Base de datos de texto](#)
- [Compuesto CSV](#)
- [Archivos binarios](#)
- [Datos heredados](#)
- [Archivos de comandos](#)
- [Datos XML](#)
- [Datos JSON](#)

## 8.1. Archivos binarios

Los archivos binarios contienen texto sin formato y caracteres binarios, como códigos de control (caracteres de ASCII código 32). El [Filtros de datos no estructurados](#) admite caracteres binarios. Puede usar caracteres binarios para definir posiciones de campos y también para valores de campos.

Un ejemplo típico sería la exportación de datos desde el sistema heredado, en el que los datos para cada etiqueta están delimitados con un carácter de Avance de página <FF>.

### 8.1.1. Ejemplo

En este caso, el desencadenador Automation captura la secuencia de impresión. La sección de datos resaltada en amarillo debe extraerse desde la secuencia y enviarse a una impresora diferente. El filtro se configura para buscar <FF> como posición final del campo.

```

<ESC>%-12345X@PJL USTATUSOFF
@PJL INFO STATUS
@PJL USTATUS DEVICE=ON
<ESC>%-12345X<ESC>%-12345X

^^02^L
^^02^00270
D11
H15
PE
SE
Q0001
131100000300070001-001-001
1e42055007500500001001019
1322000001502859
W
E
<FF><ESC>%-12345X<ESC>%-12345X@PJL USTATUSOFF
<ESC>%-12345X

```

## 8.2. Archivos de comandos

Los archivos de comandos son archivos de texto sin formato que contienen comandos que se ejecutarán uno a la vez de arriba hacia abajo. NiceLabel Automation admite archivos de comandos nativos, así como archivos de comandos SAP XML y Oracle. Para obtener más información, consulte el tema [Especificaciones de archivos de comandos](#), [Especificaciones de XML de Oracle](#) y [Todas las especificaciones de SAP XML](#).

### 8.2.1. Ejemplo

La etiqueta `label2.nlbl` se imprimirá en la impresora `CAB A3 203DPI`.

```

LABEL "label2.nlbl"
SET code="12345"
SET article="FUSILLI"
SET ean="383860026501"
SET weight="1,0 kg"
PRINTER "CAB A3 203DPI"
PRINT 1

```

## 8.3. Compuesto CSV

El compuesto CSV es un archivo de texto que contiene datos en dos estructuras: una estructura CSV estándar y un encabezado de múltiples líneas que usa una estructura no estándar. Los contenidos de un archivo CSV compuesto no pueden analizarse usando un único filtro. Para analizar datos en ambas estructuras, configure dos filtros separados:

- [Filtros de texto estructurado](#) para campos en la estructura CSV
- [Filtros de datos no estructurados](#) para campos en la estructura no estándar de encabezado

La configuración que incluye un archivo compuesto CSV, requiere dos acciones para ejecutar ambos filtros en los datos recibidos.

### 8.3.1. Ejemplo

Los datos de la línea 3 hasta el final del documento tienen estructura CSV y los analiza el filtro de texto estructurado. Los datos de las dos primeras líneas no tienen una estructura específica y los analiza el filtro de datos no estructurados.

```
OPTPEPPQPF0 NL004002 ;F75-TEP77319022891-001-001
OPT2 zg2lbppt.p 34.1.7.7 GOLF+ label
print"printer";"label";"lbl_qty";"f_logo";"f_field_1";"f_field_2";"f_field_3"
"Production01";"label.nlbl";"1";"logo-nicelabel.png";"ABCS1161P";"Post: ";"1"
"Production01";"label.nlbl";"1";"logo-nicelabel.png";"ABCS1162P";"Post: ";"2"
"Production01";"label.nlbl";"1";"logo-nicelabel.png";"ABCS1163P";"Post: ";"3"
"Production01";"label.nlbl";"1";"logo-nicelabel.png";"ABCS1164P";"Post: ";"4"
"Production01";"label.nlbl";"1";"logo-nicelabel.png";"ABCS1165P";"Post: ";"5"
```

## 8.4. Datos heredados

Los datos heredados son exportaciones no estructuradas o semiestructuradas de aplicaciones heredadas. Estas exportaciones no usan estructuras de datos CSV o XML. Para extraer datos relevantes de dichos archivos, use [Filtros de datos no estructurados](#) y defina las posiciones de los campos de interés. El filtro en Automation extrae valores de campo y los deja disponibles para la impresión en etiquetas.

### 8.4.1. Ejemplo

El archivo a continuación no sigue ninguna regla relacionada con la estructura. Cada campo debe configurarse manualmente.

```
HAWLEY      ANNIE      ER12345678 ABC      XYZ
              9876543210
```

```

PRE OP      07/11/12      F 27/06/47      St. Ken Hospital      3

G015 134 557 564 9      A- 08/11/12      LDBS F-           PB           1
G015 134 654 234 0      A- 08/11/12      LDBS F-           PB           2
G015 134 324 563 C      A- 08/11/12      LDBS F-           PB           3

```

Antibody Screen: Negative

Store Sample :

SAMPLE VALID FOR 24 HOURS, NO TRANSFUSION HISTORY SUPPLIED

```

07/11/12      B,31.0001245.E      O Rh(D) Pos      PHO
                RLUH      BT

```

## 8.5. Base de datos de texto

Base de datos de texto es un alias de archivo de texto con campos estructurados, como CSV (archivo separado con comas) o de archivo de texto con campos de ancho fijo. En ambos casos, puede hacer clic en el botón **Importar estructura de datos** y seguir el asistente para importar los campos. Si tiene un archivo de datos con estructura delimitada y el número de campos varía de una copia a la otra, puede habilitar la función **Estructura dinámica** y dejar que NiceLabel Automation manipule la extracción y la asignación de datos a las variables automáticamente. Para obtener más información, consulte el tema [Habilitar estructura dinámica](#).

### 8.5.1. Ejemplo

- **Archivo con campos delimitados:** La primera línea del archivo puede tener nombres de campo que el filtro puede importar.

```

Product_ID;Code_EAN;Product_desc;Package
CAS006;8021228110014;CASONCELLI ALLA CARNE 250G;6
PAS501;8021228310001;BIGOLI 250G;6
PAS502GI;8021228310018;TAGLIATELLE 250G;6
PAS503GI;8021228310025;TAGLIOLINI 250G;6
PAS504;8021228310032;CAPELLI D'ANGELO 250G;6

```

- **Archivo con campos de ancho fijo:** Los archivos pueden contener un número fijo de caracteres.

```

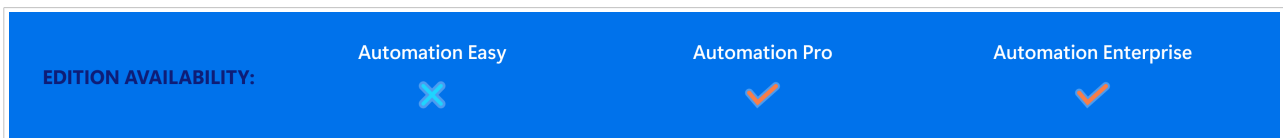
CAS006      8021228110014 CASONCELLI ALLA CARNE 250G 6
PAS501      8021228310001 BIGOLI 250G           6

```



PAS502GI	8021228310018	TAGLIATELLE	250G	6
PAS503GI	8021228310025	TAGLIOLINI	250G	6
PAS504	8021228310032	CAPELLI D'ANGELO	250G	6

## 8.6. Datos XML



XML significa lenguaje de marcado extensible. Las etiquetas XML no están predefinidas, puede definir sus propias etiquetas para describir los datos. XML es autodescriptivo.

La estructura XML está definida por los elementos, los atributos (y sus valores) y el texto (texto de elementos).

### 8.6.1. Ejemplos

#### XML de Oracle

El procesamiento de XML de Oracle es una parte integral del software. No es necesario configurar filtros para extraer datos, tan solo se debe ejecutar la acción integrada [Ejecutar archivo de comando XML de Oracle](#). Para obtener más información sobre la estructura de XML, consulte la sección [Especificaciones de XML de Oracle](#).

```
<?xml version="1.0" standalone="no"?>
<labels _FORMAT="case.nlbl" _PRINTERNAME="Production01" _QUANTITY="1">
  <label>
    <variable name="CASEID">0000000123</variable>
    <variable name="CARTONTYPE"/>
    <variable name="ORDERKEY">0000000534</variable>
    <variable name="BUYERPO"/>
    <variable name="ROUTE"></variable>
    <variable name="CONTAINERDETAILID">0000004212</variable>
    <variable name="SERIALREFERENCE">0</variable>
    <variable name="FILTERVALUE">0</variable>
    <variable name="INDICATORDIGIT">0</variable>
    <variable name="DATE">11/19/2012 10:59:03</variable>
  </label>
</labels>
```

## XML general

Si la estructura XML no tiene soporte nativo en el software, defina el filtro XML y las reglas de exportación de datos. Para obtener más información, consulte el tema [Comprender los filtros](#).

```
<?xml version="1.0" encoding="utf-8"?>
<asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
  <asx:values>
    <NICELABEL_JOB>
      <TIMESTAMP>20130221100527.788134</TIMESTAMP>
      <USER>PGRI</USER>
      <IT_LABEL_DATA>
        <LBL_NAME>goods_receipt.nlbl</LBL_NAME>
        <LBL_PRINTER>Production01</LBL_PRINTER>
        <LBL_QUANTITY>1</LBL_QUANTITY>
        <MAKTX>MASS ONE</MAKTX>
        <MATNR>28345</MATNR>
        <MEINS>KG</MEINS>
        <WDATU>19.01.2012</WDATU>
        <QUANTITY>1</QUANTITY>
        <EXIDV>012345678901234560</EXIDV>
      </IT_LABEL_DATA>
    </NICELABEL_JOB>
  </asx:values>
</asx:abap>
```

## NiceLabel XML




El procesamiento de XML de NiceLabel es una parte integral del software. No es necesario configurar filtros para extraer datos, tan solo se debe ejecutar la acción integrada [Ejecutar archivo de comandos XML](#). Para obtener más información sobre la estructura de XML, consulte la sección [Archivo de comando XML](#).

```
<nice_commands>
  <label name="label1.nlbl">

    <session_print_job printer="CAB A3 203DPI" skip=0 job_name="job name
1" print_to_file="filename 1">
      <session quantity="10">
        <variable name="variable name 1" >variable value 1</variable>
      </session>
    </session_print_job>

    <print_job printer="Zebra R-402" quantity="10" skip=0
identical_copies=1 number_of_sets=1 job_name="job name 2"
print_to_file="filename 2">
      <variable name="variable1" >1</variable>
      <variable name="variable2" >2</variable>
      <variable name="variable3" >3</variable>
    </print_job>
  </label>
</nice_commands>
```

## 8.7. Datos JSON

EDITION AVAILABILITY:	Automation Easy 	Automation Pro 	Automation Enterprise 
-----------------------	--	---	--

JavaScript Object Notation (JSON) es un formato de archivo estándar abierto. JSON utiliza texto en lenguaje natural para transmitir objetos de datos que consisten en pares de nombre-valor y tipos de datos de matriz (o cualquier otro valor serializable). JSON es un formato de datos muy común que se utiliza para la comunicación asíncrona entre navegador y servidor, incluso como reemplazo de XML.

Existen varios recursos en línea que describen las similitudes y diferencias entre JSON y XML. La siguiente tabla describe un aspecto de ellos:

JSON	XML
Es la notación de objetos de JavaScript	Es un lenguaje de marcado extensible

JSON	XML
Está basado en el lenguaje JavaScript.	Se deriva de SGML.
Es una forma de representar objetos.	Es un lenguaje de marcado y utiliza la estructura de etiquetas para representar elementos de datos.
No proporciona ningún soporte para espacios de nombres.	Admite espacios de nombres.
Admite matriz.	No admite matriz.
Los archivos son muy fáciles de leer en comparación con XML.	Los documentos son relativamente difíciles de leer e interpretar.
No usa etiqueta final.	Tiene etiquetas de inicio y final.
Es menos seguro.	Es más seguro que JSON.
No admite comentarios.	Admite comentarios.
Solo admite codificación UTF-8.	Admite varias codificaciones.

Fuente: <https://www.geeksforgeeks.org/difference-between-json-and-xml/>

## Ejemplos

```
{
  "DELIVERYNOTE": {
    "LIST_CUSTOMER_INFO": {
      "CUSTOMER_INFO": {
        "CUSTOMER_NAME": "Customer A",
        "CUSTOMER_STREET_ADDRESS": "Test St",
        "CUSTOMER_POST_ADDRESS": "1234, Test City",
        "CUSTOMER_NUMBER": "1234",
        "CURRENCY": "EUR",
        "DELIVERY_METHOD": "Express delivery",
        "EDI_INFORMATION": "EDI",
        "ORDER_TYPE": "CSO",
        "ORDER_NUMBER": "123",
        "LIST_ITEM": {
          "ITEM": [
            {
              "ARTICLE_NUMBER": "0001",
              "ARTICLE_NAME": "Collins Complete Woodworker's Manual",
              "PRICE": "23.3"
            },
            {
              "ARTICLE_NUMBER": "0002",
              "ARTICLE_NAME": "Be Careful What You Wish For (Clifton
Chronicles)",
              "PRICE": "16.6"
            },
            {
              "ARTICLE_NUMBER": "0003",
              "ARTICLE_NAME": "Mockingjay (part III of Hunger Games Trilogy)",
              "PRICE": "25.0"
            }
          ]
        }
      }
    }
  }
}
```

```
{
  "NICELABEL_JOB": {
    "TIMESTAMP": "20130221100527.788134",
    "USER": "PGRI",
    "IT_LABEL_DATA": {
      "LBL_NAME": "goods_receipt.nlbl",
      "LBL_PRINTER": "Production01",
    }
  }
}
```

```
"LBL_QUANTITY": "1",  
"MAKTX": "MASS ONE",  
"MATNR": "28345",  
"MEINS": "KG",  
"WDATU": "19.01.2012",  
"QUANTITY": "1",  
"EXIDV": "012345678901234560"  
}  
}  
}  
}
```

# 9. Referencia y resolución de problemas

## 9.1. Tipos de archivo de comando

### 9.1.1. Especificaciones de archivos de comandos

Los de comandos contienen instrucciones para el proceso de impresión. Estas instrucciones se expresan con NiceLabel comandos. Los comandos se ejecutan uno a la vez desde el comienzo hasta el final del archivo. Los archivos admiten el formato Unicode, de modo que puede incluir contenido multilingüe. Los archivos de comandos tienen tres tipos diferentes.

### 9.1.2. Archivo de comando CSV

Los comandos disponibles en los archivos de comando CSV forman un subconjunto de comandos de NiceLabel. Los siguientes comandos están disponibles: **LABEL**, **SET**, **PORT**, **PRINTER** y **PRINT**.

CSV significa valores separados por comas. El archivo CSV es un archivo de texto en el que los valores están delimitados por coma (,). Este archivo de texto puede tener valores Unicode (importantes para los datos en múltiples idiomas). Cada línea en el archivo de comandos CSV tiene los comandos para una acción de impresión de una etiqueta.

La primera fila del archivo de comandos CSV debe tener los comandos y los nombres de las variables. El orden de los comandos y los nombres no es importante, pero todos los registros en la misma secuencia de datos deben tener la misma estructura. Los pares **nombre-valor** de las variables se extraen automáticamente y se envían a la etiqueta de referencia. Si la variable con el nombre de CSV no existen en la etiqueta, no se muestran ningún mensaje de error.

#### 9.1.2.1. Ejemplo de Archivo de comando CSV

El ejemplo presenta la vista estructural de los campos que puede usar en el archivo de comando CSV.

```
@Label,@Printer,@Quantity,@Skip,@IdenticalCopies,NumberOfSets,@Port,Product_ID,
Product_Name
label1.nlbl, CAB A3 203 DPI, 100, , , , , 100A, Product 1
label2.nlbl, Zebra R-402, 20, , , , , 200A, Product 2
```

#### Especificación de comandos CSV

Los comandos de la primera línea de datos deben expresarse con el carácter (@). Los campos sin @ al principio son los nombres de las variables. Estos campos se extraen con sus valores con formato de pares **nombre-valor**.

- **@Label:** Especifica el nombre de la etiqueta que se usará. Es una buena práctica incluir el nombre de archivo y la ruta de acceso de la etiqueta. Asegúrese de que el usuario del servicio pueda acceder al archivo. Para obtener más información, consulte la sección [Acceso a recursos de red compartidos](#) en la guía del usuario NiceLabel Automation. Campo obligatorio.
- **@Printer:** Especifica la impresora que se usará. Anula la impresora definida en la etiqueta. Asegúrese de que el usuario del servicio pueda acceder a la impresora. Para obtener más información, consulte el tema [Acceso a recursos de red compartidos](#). Campo opcional.
- **@Quantity:** Especifica el número de etiquetas para imprimir. Valores posibles: valor numérico, VARIABLE o ILIMITADA. Para obtener más información, consulte el tema [Imprimir etiqueta](#). Campo obligatorio.
- **@Skip:** Especifica el número de etiquetas que se deben omitir al principio de la primera página impresa. Esta función es útil si quiere reutilizar la hoja de etiquetas parcialmente impresa. Campo opcional.
- **@IdenticalCopies:** Especifica el número de copias de cada etiqueta que se deben imprimir. Esta función es útil cuando se imprimen etiquetas con datos de bases de datos o cuando se usan contadores, y necesita copias de etiquetas. Campo opcional.
- **@NumberOfSets:** Especifica el número de veces que debe repetirse el proceso de impresión. Cada conjunto de etiquetas equivale a una única repetición del proceso de impresión. Campo opcional.
- **@Port:** Especifica el nombre del puerto para la impresora. Puede anular el puerto predeterminado en el controlador de la impresora. También puede usarlo para redireccionar la impresión al archivo. Campo opcional.
- **Otros nombres de campo:** El resto de los campos definen los nombres de las variables a partir de la etiqueta. El contenido de los campos se guarda en la variable del mismo nombre que su valor.

### 9.1.3. Archivo de comando JOB

El archivo de comandos JOB es un archivo de texto con comandos NiceLabel. Los comandos se ejecutan en orden descendente. Los comandos comienzan generalmente con LABEL (para abrir la etiqueta), luego SET (para configurar el valor de la variable) y, por último, PRINT (para imprimir la etiqueta). Para obtener más información sobre los comandos disponibles, consulte el tema [Usar comandos personalizados](#).

#### 9.1.3.1. Ejemplo de Archivo de comando JOB

El archivo JOB abre la etiqueta `label2.n1b1`, configura valores de variable e imprime una única etiqueta. Debido a que no se usa un comando PRINTER para redirigir la impresión, la etiqueta se imprimirá usando el nombre de la impresora definido en la etiqueta.

```
LABEL "label2.n1b1"  
SET code="12345"
```



```
SET article="FUSILLI"  
SET ean="383860026501"  
SET weight="1,0 kg"  
PRINT 1
```

## 9.1.4. Archivo de comando XML

Los comandos disponibles en los archivos de comando XML son un subconjunto de comandos NiceLabel. Los siguientes comandos están disponibles: **LOGIN**, **LABEL**, **SET**, **PORT**, **PRINTER**, **SESSIONEND**, **SESSIONSTART** y **SESSIONPRINT**. La sintaxis para estos comandos difiere en cierta medida si se usan en un archivo XML.

El elemento de raíz en el archivo de comandos XML es `<Nice_Commands>`. El siguiente elemento que debe seguir es `<Label>`. Este elemento especifica la etiqueta que debe usarse.

Para comenzar la impresión de etiquetas existen dos métodos: imprimir las etiquetas normalmente usando el elemento `<Print_Job>` o imprimir las etiquetas en la sesión usando el elemento `<Session_Print_Job>`. También puede cambiar la impresora con la cual se imprimirán las etiquetas, y puede configurar el valor de variable.

### 9.1.4.1. Ejemplo de Archivo de comando XML

El ejemplo a continuación brinda una descripción general estructural de los elementos y sus atributos como pueden usarse en un archivo de comando XML.

```
<nice_commands>  
  <label name="label1.nlbl">  
  
    <session_print_job printer="CAB A3 203DPI" skip=0 job_name="job name  
1" print_to_file="filename 1">  
      <session quantity="10">  
        <variable name="variable name 1" >variable value 1</variable>  
      </session>  
    </session_print_job>  
  
    <print_job printer="Zebra R-402" quantity="10" skip=0  
identical_copies=1 number_of_sets="1" job_name="job name 2"  
print_to_file="filename 2">  
      <variable name="variable1" >1</variable>  
      <variable name="variable2" >2</variable>  
      <variable name="variable3" >3</variable>  
    </print_job>  
  </label>  
</nice_commands>
```

## Especificación del archivo de comando XML

Esta sección presenta la descripción de la estructura de los archivos de comandos XML. Hay varios elementos que contienen atributos. Algunos atributos son obligatorios, otros son opcionales. Algunos atributos pueden tener valores predefinidos únicamente y, para otros, se pueden especificar valores personalizados.

- **<Nice\_Commands>**: este es un elemento de raíz.
- **<Label>**: Especifica el archivo de etiquetas que se abre. Si la etiqueta ya está abierta, no se volverá a abrir. Se debe poder acceder al archivo de etiqueta desde esta computadora. Para obtener más información, consulte el tema [Acceso a recursos de red compartidos](#). Este elemento puede aparecer varias veces dentro del archivo de comandos.
  - **Nombre**: Este atributo tiene el nombre de la etiqueta. Puede incluir la ruta de acceso junto con el nombre de la etiqueta. Obligatorio.
- **<Print\_Job>**: Elemento que contiene los datos para un único trabajo de etiquetas. Este elemento puede aparecer varias veces dentro del archivo de comandos.
  - **Impresora**:: Use este atributo para reemplazar la impresora definida en la etiqueta. Se debe poder acceder a la impresora desde esta computadora. Para obtener más información, consulte el tema [Acceso a recursos de red compartidos](#). Opcional.
  - **Cantidad**: Use este atributo para especificar el número de etiquetas que se imprimirán.. Valores posibles: valor numérico, VARIABLE o ILIMITADA. Para obtener más información sobre los parámetros, consulte el tema [Imprimir etiqueta](#). Obligatorio.
  - **Skip**: Use este atributo para especificar cuántas etiquetas se omitirán al comienzo. Esta función es útil para imprimir hojas de etiquetas con una impresora láser, pero las hojas están parcialmente impresas. Para obtener más información, consulte el tema [Imprimir etiqueta](#). Opcional.
  - **Job\_name**: Use este atributo para especificar el nombre del archivo de trabajo. El nombre especificado aparece en el administrador de trabajos de impresión. Para obtener más información, consulte el tema [Configurar nombre de trabajo de impresión](#). Opcional.
  - **Print\_to\_file**: Use este atributo para especificar el nombre del archivo en el que quiera guardar los comandos de la impresora. Para obtener más información, consulte el tema [Redireccionar impresión a archivo](#). Opcional.
  - **Identical\_copies**: Use este atributo para especificar el número de copias que necesita para cada etiqueta. Para obtener más información, consulte el tema [Imprimir etiqueta](#).
- **<Session\_Print\_Job>**: Elemento que contiene los comandos y los datos para una o más sesiones. Este elemento puede tener uno o más elementos **<Session>**. Tiene en cuenta las reglas de impresión de la sesión. Puede usar este elemento varias veces dentro del archivo de comandos. Busque los atributos disponibles para el elemento **<Print\_Job>**. Todos son válidos, tan solo no puede usar el atributo de cantidad. Consulte la descripción del elemento **<Session>** para buscar cómo especificar la cantidad de etiquetas en la impresión de la sesión.
- **<Session>**: Elemento que contiene los datos para una sesión. Al imprimir en la sesión, todas las etiquetas se codifican en un solo trabajo de impresión y se envían a la impresora.

- **Quantity:** Use este atributo para especificar el número de etiquetas que se imprimirán.. Valores posibles: valor numérico, VARIABLE de cadena o cadena ILIMITADA. Para obtener más información sobre los parámetros, consulte el tema [Imprimir etiqueta](#). Obligatorio.
- **<Variable>:** Elemento que configura el valor de las variables de la etiqueta. Este elemento puede aparecer varias veces dentro del archivo de comandos.
  - **Nombre:** Este atributo tiene el nombre de la variable. Obligatorio.

## Definición de esquema XML (XSD) para archivo de comandos XML

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema targetNamespace="http://tempuri.org/XMLSchema.xsd"
elementFormDefault="qualified" xmlns="http://tempuri.org/XMLSchema.xsd"
xmlns:mstns="http://tempuri.org/XMLSchema.xsd" xmlns:xs="http://www.w3.org/
2001/XMLSchema">
  <xs:element name="nice_commands">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="label" maxOccurs="unbounded" minOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="print_job" maxOccurs="unbounded"
minOccurs="0">
                <xs:complexType>
                  <xs:sequence>
                    <xs:element name="database"
maxOccurs="unbounded" minOccurs="0">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension
base="xs:string">
                            <xs:attribute
name="name" type="xs:string" use="required" />
                          </xs:extension>
                        </xs:simpleContent>
                      </xs:complexType>
                    </xs:element>
                  <xs:element name="table"
maxOccurs="unbounded" minOccurs="0">
                      <xs:complexType>
                        <xs:simpleContent>
                          <xs:extension
base="xs:string">
                            <xs:attribute
name="name" type="xs:string" use="required" />
                          </xs:extension>
                        </xs:extension>
                      </xs:complexType>
                    </xs:element>
                  </xs:sequence>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

        </xs:simpleContent>
    </xs:complexType>
</xs:element>
<xs:element name="variable"
maxOccurs="unbounded" minOccurs="0">
    <xs:complexType>
        <xs:simpleContent>
            <xs:extension
base="xs:string">
                <xs:attribute
name="name" type="xs:string" use="required" />
            </xs:extension>
        </xs:simpleContent>
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="quantity"
type="xs:string" use="required" />
<xs:attribute name="printer"
type="xs:string" use="optional" />
<xs:attribute name="skip"
type="xs:integer" use="optional" />
<xs:attribute name="identical_copies"
type="xs:integer" use="optional" />
<xs:attribute name="number_of_sets"
type="xs:integer" use="optional" />
<xs:attribute name="job_name"
type="xs:string" use="optional" />
<xs:attribute name="print_to_file"
type="xs:string" use="optional" />
<xs:attribute name="print_to_file_append"
type="xs:boolean" use="optional" />
<xs:attribute name="clear_variable_values"
type="xs:boolean" use="optional" />
    </xs:complexType>
</xs:element>
<xs:element name="session_print_job"
maxOccurs="unbounded" minOccurs="0">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="database"
maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension
base="xs:string">

```

```

        <xs:attribute
name="name" type="xs:string" use="required" />
        </xs:extension>
        </xs:simpleContent>
        </xs:complexType>
</xs:element>
<xs:element name="table"
maxOccurs="unbounded" minOccurs="0">
        <xs:complexType>
        <xs:simpleContent>
        <xs:extension
base="xs:string">
                <xs:attribute
name="name" type="xs:string" use="required" />
                </xs:extension>
                </xs:simpleContent>
                </xs:complexType>
</xs:element>
<xs:element name="session"
minOccurs="1" maxOccurs="unbounded">
        <xs:complexType>
        <xs:sequence>
        <xs:element
name="variable" minOccurs="0" maxOccurs="unbounded">
                <xs:complexType>
                <xs:simpleContent>
                <xs:extension
base="xs:string">
                        <xs:attribute name="name" type="xs:string" use="required"
/>
                        </
xs:extension>
                                </xs:simpleContent>
                                </xs:complexType>
                                </xs:element>
                                </xs:sequence>
                                <xs:attribute name="quantity"
type="xs:string" use="required" />
                                </xs:complexType>
                                </xs:element>
                                </xs:sequence>
                                <xs:attribute name="printer"
type="xs:string" use="optional" />
                                <xs:attribute name="skip"
type="xs:integer" use="optional" />

```

```

        <xs:attribute name="job_name"
type="xs:string" use="optional" />
        <xs:attribute name="print_to_file"
type="xs:string" use="optional" />
        <xs:attribute name="print_to_file_append"
type="xs:boolean" use="optional" />
        <xs:attribute name="clear_variable_values"
type="xs:boolean" use="optional" />
    </xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="name" type="xs:string"
use="required" />
<xs:attribute name="close" type="xs:boolean"
use="optional" />
<xs:attribute name="clear_variable_values"
type="xs:boolean" use="optional" />
</xs:complexType>
</xs:element>
</xs:sequence>
<xs:attribute name="quit" type="xs:boolean" use="required" />
</xs:complexType>
</xs:element>
</xs:schema>

```

## 9.1.5. Especificaciones de XML de Oracle

Oracle define el formato XML para que se pueda comprender, analizar y luego imprimir el contenido XML en las etiquetas. La Definición del tipo de documento XML (DTD) define las etiquetas XML que se usarán en un archivo XML. Oracle generará archivos XML según esta DTD y el software de terceros traducirá los archivos XML según esta DTD.

Para ejecutar un archivo de comando XML de Oracle, use la acción [Ejecutar archivo de comando XML de Oracle](#).

### 9.1.5.1. XML DTD

A continuación se muestra la DTD de XML que se usa para formar el XML para los formatos de XML sincrónicos y asincrónicos. La DTD define los elementos que se utilizan en el archivo XML, una lista de sus atributos y los elementos del nivel siguiente.

```

<!ELEMENT labels (label)*>
<!ATTLIST labels _FORMAT CDATA #IMPLIED>
<!ATTLIST labels _JOBNAME CDATA #IMPLIED>

```

```

<!ATTLIST labels _QUANTITY CDATA #IMPLIED>
<!ATTLIST labels _PRINTERNAME CDATA #IMPLIED>
<!ELEMENT label (variable)*>
<!ATTLIST label _FORMAT CDATA #IMPLIED>
<!ATTLIST label _JOBNAME CDATA #IMPLIED>
<!ATTLIST label _QUANTITY CDATA #IMPLIED>

```

### 9.1.5.2. Ejemplo de XML de Oracle

Este es el XML de Oracle que brinda datos para una etiqueta (hay solo un elemento <label>).

```

<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!DOCTYPE labels SYSTEM "label.dtd">
<labels _FORMAT = "Serial.nlbl" _QUANTITY="1" _PRINTERNAME="" _JOBNAME="Serial">
  <label>
    <variable name= "item">O Ring</variable>
    <variable name= "revision">V1</variable>
    <variable name= "lot">123</variable>
    <variable name= "serial_number">12345</variable>
    <variable name= "lot_status">123</variable>
    <variable name= "serial_number_status">Active</variable>
    <variable name= "organization">A1</variable>
  </label>
</labels>

```

Al ejecutar este archivo XML de Oracle de ejemplo, se imprime la etiqueta **serial.nlbl** con los siguientes valores de variables.

Nombre de variable	Valor de variable
Elemento	Anillo tórico
revisión	V1
lote	123
núm_de_serie	12345
estado_de_lote	123
estado_núm_de_serie	Activo
organización	A1

Habrà 1 copia impresa de la etiqueta con el nombre **serial** en el administrador de trabajos de impresión. El nombre de la impresora no se especifica en el archivo XML, de modo que la etiqueta se imprimirà en la impresora definida en la plantilla de la etiqueta.

## 9.1.6. Todas las especificaciones de SAP XML

NiceLabel Automation se puede presentar como un controlador de dispositivos RFID, capaz de codificar etiquetas RFID e imprimir etiquetas. Para obtener más información sobre Todas las especificaciones de SAP XML, consulte el documento **Interfaz de controlador de dispositivo de infraestructura de ID automático de SAP** en la página Web de SAP.

Para ejecutar dicho archivo de comando, use la acción Ejecutar Todas las especificaciones de SAP XML.

### 9.1.6.1. Ejemplo de SAP Todo XML

Este es todo XML de SAP que brinda datos para una etiqueta (hay solo un elemento `<label>`).

```
<?xml version="1.0" encoding="UTF-8"?>
<Command xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Command.xsd">
  <WriteTagData readerID="DEVICE ID">
    <Item>
      <FieldList format="c:\SAP Demo\SAP label.nlbl"
jobName="Writer_Device20040929165746" quantity="1">
        <Field name="EPC">00037000657330</Field>
        <Field name="EPC_TYPE">SGTIN-96</Field>
        <Field name="EPC_URN">urn:autoid:tag:sgtin:3.5.0037000.065774.8</
Field>
        <Field name="PRODUCT">Product</Field>
        <Field name="PRODUCT_DESCRIPTION">Product description</Field>
      </FieldList>
    </Item>
  </WriteTagData>
</Command>
```

Al ejecutar este archivo SAP Todo XML de ejemplo, se imprimirá la etiqueta `c:\SAP Demo\SAP label.nlbl` con los siguientes valores de variables.

Nombre de variable	Valor de variable
EPC	00037000657330
TIPO_EPC	SGTIN-96
EPC	urn:idautomático:etiqueta:sgtin:3.5.0037000.065774.8
PRODUCTO	Producto
DESCRIPCIÓN_PRODUCTO	Descripción de producto

Habrá 1 copia impresa de la etiqueta con el nombre `Writer_Device2004092916574` en el administrador de trabajos de impresión. El nombre de la impresora no se especifica en el archivo XML, de modo que la etiqueta se imprimirá en la impresora definida en la plantilla de la etiqueta.



## 9.1.7. Archivo de comandos PAS

El archivo de comandos PAS es un archivo de texto que contiene comandos de impresión de Loftware LPS. Los comandos se ejecutan en orden de arriba abajo. Cada archivo PAS debe empezar con el comando \*FORMAT (abrir etiqueta) y finalizar con el comando \*PRINTLABEL (imprimir etiqueta).

El archivo de comandos PAS se puede ejecutar con la acción [Ejecutar archivo de comando](#).



### NOTA

Si ha realizado la migración de Loftware a NiceLabel, puede usar sus archivos de comandos PAS existentes con las plantillas de etiqueta de NiceLabel (.nlbl). El comando \*FORMAT verifica si el archivo .nlbl existe con el mismo nombre como .lwl.

### 9.1.7.1. Ejemplo de archivo de comandos PAS

```
*FORMAT, \\127.0.0.1\labelcomputer$\labels\ML2193.lwl
*JOBNAME, PastaPallet002
*QUANTITY, 1
*PRINTERNUMBER, 1
Text0000, New Data
Text0001, New Data
*PRINTLABEL
```



### NOTA

Si desea utilizar el comando \*PRINTERNUMBER, debe copiar el archivo printr32.ini de LPS a c:\Program Files\NiceLabel\NiceLabel 10\NiceLabel\bin.net\Configuration.

Para obtener más detalles sobre los comandos PAS, consulte la Guía del usuario de Loftware LPS.

## 9.2. Comandos personalizados




### 9.2.1. Usar comandos personalizados

Los comandos NiceLabel se usan en archivos de comandos para controlar la impresión de etiquetas. NiceLabel Automation ejecuta los comandos dentro de los archivos de comandos en orden descendente. Para obtener más información, consulte el tema [Especificaciones de archivos de comandos](#).

Se puede usar el comando personalizado específico cuando esté disponible en su producto NiceLabel Automation como una acción.

## Ejemplo

Es posible usar el comando **SETPRINTPARAM** si puede seleccionar la acción **Establecer parámetro de impresión** (disponible con niveles de producto Pro y Enterprise).

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
			

## NiceLabel Especificación de comandos

### COMMENT

```
;
```

Al desarrollar un archivo de comando, es una buena práctica documentar los comandos. De este modo, será más fácil descodificar qué secuencia realmente desempeña cuando observe el código después de un tiempo. Use punto y coma (;) al comienzo de la línea. Todo lo que esté detrás del punto y coma se tratará como comentario y no se procesará.

### CLEARVARIABLEVALUES

```
CLEARVARIABLEVALUES
```

Este comando restablece el valor predeterminado de los valores de las variables.

### CREATEFILE

```
CREATEFILE <file name> [, <contents>]
```

Este comando crea un archivo de texto. Use el archivo de texto para indicar a una aplicación de terceros que comenzó o finalizó el proceso de impresión, según la ubicación en la que incluyó el comando. Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte el tema [Acceso a recursos de red compartidos](#).

### DELETEFILE

```
DELETEFILE <file name>
```

Elimina el archivo especificado. Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte el tema [Acceso a recursos de red compartidos](#).

### EXPORTLABEL

```
EXPORTLABEL ExportFileName [, ExportVariant]
```

Este comando se implementa para automatizar el comando "Exportar a impresora" que está disponible en el diseñador de etiquetas. La etiqueta se exporta directamente a la impresora y se almacena en la memoria para la impresión sin conexión. El usuario recupera la etiqueta con el teclado en la impresora o enviando un archivo de comando a la impresora. La misma funcionalidad también está disponible con la acción [Almacenar etiqueta en impresora](#).



## NOTA

Para especificar qué etiqueta se exportará, use primero el comando **LABEL**.

- **ExportFileName:** Este parámetro es obligatorio y define el nombre de archivo de comandos de impresora generados.
- **ExportVariant:** Algunas impresoras admiten múltiples variantes de exportación. Al exportar en forma manual, el usuario puede seleccionar la variante de exportación en el cuadro de diálogo. Con el comando EXPORTLABEL, debe especificar qué variante de exportación quiere usar. Las variantes se ven en el diseñador de etiquetas una vez que habilita el modo de impresión Almacenar/Recuperar. La primera variante de la lista tiene el valor 0. La segunda variante tiene el valor 1, etc. Si no especifica algún tipo de variante, se usa en forma predeterminada el valor 0.

Para obtener más información sobre impresión en conexión, consulte la sección [Usar modo de impresión almacenar/recuperar](#).

## IGNOREERROR

```
IGNOREERROR <on> [, <off>]
```

Este comando especifica que el error que aparece en el archivo JOB no finalizará el proceso de impresión si se generan los siguientes errores:

- Uso de nombre de variable incorrecto.
- Valor incorrecto enviado a la variable.
- No existe / no está accesible la etiqueta.
- No existe / no está accesible la impresora.

## LABEL

```
LABEL <label name> [, <printer_name>]
```

El comando abre la etiqueta para imprimir. Si la etiqueta ya está cargada, no se volverá a abrir. Puede incluir el nombre de la ruta de acceso. Escriba el nombre de la etiqueta entre comillas dobles si el nombre o la ruta de acceso tiene espacios. Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte el tema [Acceso a recursos de red compartidos](#).

El `printer_name` opcional especifica la impresora, para la cual se abrirá la etiqueta. Use esta configuración si quiere anular el nombre de la impresora que se guardó en la plantilla de la etiqueta. Si el controlador para el nombre de la impresora provisto no está instalado o disponible, el comando arroja un error.

## MESSAGEBOX

```
MESSAGEBOX <message> [,<caption>]
```

Este comando registra el **mensaje** personalizado en el registro de desencadenadores. Si el mensaje tiene caracteres de espacio o comas, debe escribir el texto entre comillas dobles (").

## PORT

```
PORT <file name> [ , APPEND]
```

Este comando anula el puerto definido en el controlador de la impresora y redirige la impresión a un archivo. Si la ruta de acceso o el nombre de archivo tiene espacios, encierre el valor entre comillas dobles ("). Utilice la sintaxis UNC para los recursos de red. Para obtener más información, consulte el tema [Acceso a recursos de red compartidos](#).

El parámetro **APPEND** es opcional. El archivo se sobrescribe en forma predeterminada. Use este parámetro para anexar datos en el archivo existente.

Una vez que usa un comando PORT en el archivo JOB, seguirá siendo válido hasta el siguiente comando PORT, o hasta el final del archivo (lo que ocurra primero). Si usa el comando PRINTER después de ejecutar el comando PORT, la configuración PORT sobrescribe el puerto definido para la impresora seleccionada. Si quiere usar el puerto real definido para la impresora seleccionada, debe usar otro comando PORT con un valor vacío, como `PORT = ""`.

## PRINT

```
PRINT <quantity> [,<skip> [,<identical label copies> [ ,number of label sets]]]
```

Este comando comienza el proceso de impresión.

- **Quantity:** especifica la cantidad de etiquetas impresas.
  - **<number>:** El número específico de etiquetas que se imprimen.
  - **VARIABLE:** Especifica que alguna variable de la etiqueta está definida como *cantidad variable* y que contiene la cantidad de etiquetas a imprimir. La etiqueta determinará cuántas etiquetas se imprimirán.
  - **UNLIMITED:** Si usa una base de datos para adquirir valores de objetos, la impresión ilimitada imprimirá tantas etiquetas como registros haya en la base de datos. Si no usa una base de datos, se imprimirá el número máximo de etiquetas que la impresora térmica admita internamente.
- **Skip:** Especifica el número de etiquetas que quiere omitir en la primera página. El parámetro se usa para imprimir etiquetas en hojas de papel. Si ya usó parte de la página, puede reutilizar la misma hoja cambiando la ubicación de inicio de la primera etiqueta.
- **Copias de etiquetas idénticas:** Especifica cuántas copias de la misma etiqueta se deben imprimir.

- **Número de juegos de etiquetas.** Especifica el número de veces que debe repetirse el proceso de impresión completo.



## NOTA

Asegúrese de que los valores de cantidad se provean como valor numérico, no como valor de cadena. No escriba el valor entre comillas dobles.

## PRINTER

```
PRINTER <printer name>
```

Este comando anula la impresora definida en el archivo de etiqueta. Si el nombre de la impresora tiene caracteres de espacio o comas, escríbalo entre comillas dobles (").

Use el nombre de la impresora como se muestra en la línea de estado en la aplicación de diseño de etiquetas. Los nombres de las impresoras generalmente son los mismos que los de Impresoras y Faxes del Panel de control, pero no siempre. Si usa impresoras de redes, es posible que vea el nombre mostrado con la sintaxis `\\server\share`.

## PRINTJOBNAME

```
PRINTJOBNAME
```

Este comando especifica el nombre del trabajo de impresión como se muestra en el Administrador de trabajos de impresión de Windows. Si el nombre tiene caracteres de espacio o comas, debe escribir el texto entre comillas dobles (").

## SESSIONEND

```
SESSIONEND
```

Este comando cierra la secuencia de impresión. Consulte también **SESSIONSTART**.



## NOTA

`SESSIONEND` debe enviarse como el único elemento en la acción Enviar comandos personalizados. Si desea enviar comandos adicionales, utilice acciones Enviar comandos personalizados separadas.

## SESSIONPRINT

```
SESSIONPRINT <quantity> [,<skip>]
```

Este comando imprime la etiqueta de referencia actual y la agrega a la secuencia de impresión de la sesión abierta actualmente. Puede usar múltiples comandos de SESSIONPRINT uno después del otro y unir las etiquetas de referencia en una sola secuencia de impresión. La secuencia no se cierra si no se usa el comando SESSIONEND. La función de los parámetros de cantidad y omitir es la misma que la de PRINT de NiceCommand. Consulte también el comando **SESSIONSTART**.

- **Quantity:** Especifica el número de etiquetas para imprimir.
- **Skip:** Especifica el número de etiquetas que quiere omitir en la primera página. El parámetro se usa para imprimir etiquetas en hojas de papel. Si ya usó parte de la página, puede reutilizar la misma hoja cambiando la ubicación de inicio de la primera etiqueta.

## SESSIONSTART

```
SESSIONSTART
```

Este comando empieza el tipo de sesión de la impresión.

Los tres comandos relacionados con la sesión de impresión (**SESSIONSTART**, **SESSIONPRINT**, **SESSIONEND**) se usan juntos. Cuando usa el comando PRINT, cada dato de la etiqueta se envía a la impresora en un trabajo de impresión separado. Si quiere unir datos de etiquetas para múltiples etiquetas de la secuencia de impresión, debe usar los comandos de impresión de la sesión. Para ello, comience con el comando SESSIONSTART, seguido de cualquier número de comandos SESSIONPRINT. La secuencia finaliza con el comando SESSIONEND.

Use estos comandos para optimizar el proceso de impresión de etiquetas. La impresión de etiquetas que pertenecen a un único trabajo de impresión es mucho más rápida que la impresión de etiquetas utilizando múltiples trabajos de impresión.

Use las reglas a continuación para asegurarse de que la impresión de sesiones no se interrumpa:

- No puede cambiar la etiqueta dentro de una sesión.
- No puede cambiar la impresora dentro de una sesión.
- Debe establecer valores para todas las variables de la etiqueta dentro de una sesión, incluso si algunas variables tienen valores vacíos.

## SET

```
SET <name>=<value> [,<step> [,<number or repetitions>]]
```

Este comando le asigna al **name** de la variable un **value**. La variable debe definirse en la etiqueta, o se genera un error. Si la variable no está en la etiqueta, ocurre un error. **step** y **number of repetitions** son parámetros para las variables del contador. Estos parámetros especifican el valor del incremento del contador y el número de etiquetas impresas antes de que cambie el valor del contador.

Si el valor tiene caracteres de espacio o comas, debe escribir el texto entre comillas dobles ("). Consulte también **TEXTQUALIFIER**.

Si quiere asignar un valor de múltiples líneas, use `\r\n` para codificar un carácter de línea nueva. `\r` se reemplaza con CR (Retorno de carro) y `\n` se reemplaza con LF (Avance de línea).

Tenga cuidado al establecer valores para variables que brindan datos para las imágenes de la etiqueta, ya que los caracteres de barra invertida pueden reemplazarse con otros caracteres.

## Ejemplo

Si asigna el valor "c:\My Pictures\raw.jpg" a la variable, "\r" se reemplazará con el carácter CR.

## SETPRINTPARAM

```
SETPRINTPARAM <paramname> = <value>
```

Este comando le permite ajustar la configuración de la impresora antes de imprimir. Los parámetros admitidos para la configuración de la impresora (**paramname**) son:

- **PAPERBIN:** Especifica la bandeja que tiene los medios de las etiquetas. Si la impresora está equipada con más de una bandeja de papel / etiquetas, puede controlar cuál se usa para la impresión. El nombre de la bandeja debe obtenerse del controlador de la impresora.
- **PRINTSPEED:** Especifica la velocidad de impresión. Los valores aceptables varían según la impresora. Consulte los manuales de las impresoras para saber el rango exacto de valores.
- **PRINTDARKNESS:** Especifica la oscuridad / contraste de impresión. Los valores aceptables varían según la impresora. Consulte los manuales de las impresoras para saber el rango exacto de valores.
- **PRINTOFFSETX:** Especifica el desplazamiento izquierdo para todos los objetos de impresión. El valor para este parámetro debe ser numérico, positivo o negativo, con punto.
- **PRINTOFFSETY:** Especifica el desplazamiento superior para todos los objetos de impresión. El valor para este parámetro debe ser numérico, positivo o negativo, con punto.
- **PRINTERSETTINGS:** Especifica la configuración de la impresora personalizada que se aplicará al trabajo de impresión. El parámetro requiere toda la estructura DEVMODE para la impresora de destino, provista en una cadena codificada como Base64. La estructura DEVMODE tiene todos los parámetros del controlador de la impresora al mismo tiempo (velocidad, oscuridad, desplazamientos y otros). Para obtener más información, consulte el tema [Comprender la configuración de la impresora y DEVMODE](#).



### NOTA

La cadena codificada como Base64 debe escribirse entre comillas dobles (").

## TEXTQUALIFIER

```
TEXTQUALIFIER <character>
```



El calificador de texto es el carácter que incluye el valor de los datos asignados a una variable. Siempre que los datos tengan caracteres de espacio, deben incluirse con calificadores de texto. El calificador de texto es un carácter de comillas dobles ("). Debido a que las comillas dobles se usan como acceso directo para la unidad de medida de pulgadas, algunas veces es difícil pasar los datos con marcas de pulgadas en los archivos JOB. También puede usar dos comillas dobles para codificar un carácter de comillas dobles, o usar TEXTQUALIFIER.

### Ejemplo

```
TEXTQUALIFIER %  
  
SET Variable = %EPAK 12"X10 7/32"%
```

## 9.3. Acceso a recursos de red compartidos

Este tema define los pasos de las mejores prácticas para utilizar recursos de red compartidos.

- **Privilegios de usuario para el modo de servicio.** El componente de ejecución de NiceLabel Automation se ejecuta en el modo de servicio según los privilegios de acceso de herencia especificados para la cuenta de usuario. Para que NiceLabel Automation pueda abrir los archivos de etiqueta y utilizar controladores de impresoras en la cuenta de usuario asociada debe contar con suficientes privilegios. Para obtener más información, consulte el tema [Ejecutar en modo de servicio](#).
- **Notación UNC para recursos compartidos de red.** Al ingresar al archivo en una unidad de red, asegúrese de utilizar la sintaxis UNC (convención de nomenclatura universal) y no las letras de la unidad asignada. UNC es una convención de nomenclatura para especificar y mapear unidades de red. NiceLabel Automation intentará reemplazar la sintaxis de letras de la unidad con la sintaxis UNC automáticamente.

### Ejemplo

Si el archivo está disponible como `G:\Etiquetas\etiqueta.nlbl`, consulte la notación UNC `\\servidor\recursos compartidos\Etiquetas\etiqueta.nlbl` (donde la unidad G: está asignada a `\\servidor\recursos compartidos`).

- **Notación para ingresar archivos en el Centro de control.** Cuando abre el archivo en el Almacenamiento de documentos dentro del Centro de control, puede utilizar la notación HTTP `http://nombredeservidor:8080/etiqueta.lbl`, o la notación WebDAV `\\nombredeservidor@8080\DavWWWRaíz\etiqueta.lbl`.

Notas adicionales:

- La cuenta de usuario en la cual NiceLabel Automation se ejecuta el servicio también se usa para obtener archivos del Almacenamiento de documentos. Este usuario debe estar configurado en la configuración del Centro de control. Esto permite al usuario asegurarse de tener acceso a los archivos del Almacenamiento de documentos.

- El acceso WebDAV puede utilizarse únicamente con la autenticación de usuario de Windows en el Centro de control.



#### NOTA

El Almacenamiento de documentos está disponible con los productos **LMS Enterprise** y **LMS Pro**.

- **Disponibilidad de controladores de impresoras.** Para imprimir etiquetas en una impresora de red compartida, debe habilitar el controlador en el servidor en el que NiceLabel Automation está instalado. Asegúrese de que la cuenta de usuario que ejecuta el Servicio NiceLabel Automation tenga acceso al controlador de la impresora. Si la impresora de red está recién instalada en la máquina, es posible que NiceLabel Automation no la vea hasta que se reinicie el Servicio. Para permitir la notificación automática de los nuevos controladores de la impresora de red, debe habilitar la regla de entrada correspondiente en el firewall de Windows. Para obtener más información, consulte [Artículo de base de conocimiento](#).

## 9.4. Almacenamiento de documentos y versiones de archivos de configuración

Almacenamiento de documentos es una funcionalidad de NiceLabel Control Center. Hace que NiceLabel Control Center funcione como un repositorio de archivos compartidos en el servidor, donde los usuarios pueden guardar sus archivos, recuperarlos y controlar sus revisiones.

La pestaña contextual **Almacenamiento de documentos** permite realizar acciones de almacenamiento de documentos directamente desde Automation Builder. De este modo, no es necesario acceder y abrir el archivo Automation en NiceLabel Control Center.



#### NOTA

Esta pestaña contextual necesita la conexión con NiceLabel Control Center. La licencia de LMS Enterprise es obligatoria para dichas configuraciones.

El grupo **Revisiones** permite realizar las acciones de almacenamiento de documentos disponibles:

- **Desproteger:** habilita el archivo del NiceLabel Control Center almacenamiento de documentos de modo que se pueda editar. El archivo desprotegido se marca y bloquea para la edición de otro usuario. Todos los otros usuarios verán la revisión actual del archivo, mientras que el autor ya esté trabajando con un borrador nuevo.



#### NOTA

Después de abrir un documento desde el almacenamiento de documentos (**Archivo > Abrir > Almacenamiento de documentos**), el comando de edición permanece desactivado hasta que desproteja el documento.

- **Proteger:** vuelve a bloquear el archivo en el NiceLabel Control Center almacenamiento de documentos una vez que finaliza la edición. Cuando protege el archivo, la revisión del archivo aumentará de a uno. El comentario ingresado se agrega al registro de archivos.
- **Descartar desprotección:** descarta la desprotección del archivo actual y otorga acceso total al archivo a otros usuarios.



### AVISO

Si hace clic en **Descartar desprotección**, se perderán todos los cambios desde la última desprotección de archivos.

- **Almacenamiento de documentos:** abre la ubicación del almacenamiento de documentos del NiceLabel Control Center conectado.

## 9.5. Acceso a bases de datos

Siempre que NiceLabel Automation tenga que obtener datos de alguna base de datos, debe asegurarse de que el controlador de bases de datos necesario esté instalado en el sistema Windows. Los controladores de las bases de datos los proporciona la empresa que desarrolla el software de las bases de datos. El controlador que instale debe tener el mismo valor de bits que el sistema Windows. El software de NiceLabel siempre se ejecutará en el valor de bits de su sistema Windows.

### 9.5.1. Windows de 32 bits

Si tiene Windows de 32 bits, solo puede instalar controladores de bases de datos de 32 bits. Se utilizará el mismo controlador de bases de datos para configurar el desencadenador en Automation Builder y para ejecutar el desencadenador en el Servicio NiceLabel Automation. Todos los componentes NiceLabel Automation se ejecutarán como aplicaciones de 32 bits.

### 9.5.2. Windows de 64 bits

Si tiene Windows de 64 bits, puede instalar controladores de bases de datos de 32 o de 64 bits. Las aplicaciones que se ejecutan en 64 bits utilizarán controladores de bases de datos de 64 bits. Las aplicaciones que se ejecutan en 32 bits utilizarán controladores de bases de datos de 32 bits.

De manera predeterminada, el Automation Servicio se ejecuta como un proceso de 64 bits. Como tal, usará controladores de base de datos de 64 bits para conectarse a una conexión de la base de datos. Cuando los controladores de base de datos de 64 bits no están disponibles en el sistema donde se está ejecutando el Servicio Automation, la tarea de conexión de la base de datos se descargará al proceso de **NiceLabel Proxy**. Este proceso siempre se ejecuta como la aplicación de 32 bits.

## 9.6. Reemplazo de fuente automático

Puede diseñar sus plantillas de etiquetas para imprimir objetos de texto utilizando fuentes de impresora internas. Estas son las fuentes que se guardan en la memoria de su impresora. Si intenta imprimir dichas etiquetas en un tipo diferente de impresora, las fuentes internas seleccionadas podrían no estar disponibles. Es probable que la nueva impresora sea compatible con un conjunto de fuentes internas completamente distintas. El diseño de fuente puede ser parecido en ese caso, pero está disponible con un nombre diferente.

La discrepancia de fuentes también podría ocurrir si la fuente TrueType que está usando en sus etiquetas no está instalada en la computadora en la que ejecuta Designer para diseñar e imprimir etiquetas.

Designer puede configurarse de modo que reemplace automáticamente las fuentes utilizadas en la etiqueta con fuentes compatibles. En ese caso, Designer asigna y reemplaza las fuentes usando sus nombres. Cuando no se encuentre la fuente original, Designer utilizará la primera fuente de reemplazo disponible como se define en la tabla de asignación.

Si no hay fuentes de reemplazo adecuadas, Designer utiliza la fuente Arial TrueType.



### NOTA

Después de configurar el reemplazo de fuentes, cuando cambia la impresora para su etiqueta se ejecutan las reglas de asignación.

## Configuración de la asignación de fuentes

1. Abra el Explorador de archivos y navegue a la siguiente carpeta:

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10
```

2. Copie el archivo **fontmapping.def** a **fontmapping.local.def**.
3. Abra el archivo **fontmapping.local.def** en su editor de texto XML preferido.
4. Dentro del elemento **FontMappings**, cree un nuevo elemento con un nombre personalizado.
5. Dentro del nuevo elemento, cree al menos dos elementos llamados **Asignación**.
  - El valor del primer elemento llamado Asignación debe contener el nombre de la fuente original.
  - El valor del segundo elemento llamado Asignación debe contener el nombre de la fuente de reemplazo.



### NOTA

Se permiten elementos de Asignación adicionales con nuevos nombres de fuente. Si la primera fuente de reemplazo no está disponible, Designer intenta con la siguiente. Si no hay fuentes de reemplazo disponibles, se usará la fuente Arial Truetype.



### NOTA

El archivo **fontmapping.local.def** es su archivo y se conserva durante las actualizaciones. Por otro lado, **fontmapping.def** pertenece a NiceLabel y se sobrescribe durante las actualizaciones. No modifique el archivo **fontmapping.def**.




## Configuración de asignación de muestra

En este ejemplo, se definen dos asignaciones de fuente.

- La primera asignación convierte a cualquier fuente **Avery** en la fuente **Novexx** correspondiente. Por ejemplo, la fuente **Avery YT100** se reemplazará con la fuente **Novexx YT100**, la fuente **Avery 1**, con **Novexx 1**. Si la fuente Novexx no está disponible, se utilizará la fuente **Arial** Truetype.
- La segunda asignación convertirá a la fuente **Avery YT100** en **Novexx YT104**. Si esa fuente no está disponible, entonces se usará **Zebra 0**. Si esa fuente no está disponible, se utilizará **Arial** Truetype.
- La segunda asignación anula la primera.

```
<?xml version="1.0" encoding="utf-8"?>
<FontMappings>
  <AveryNovexx>
    <Mapping>Avery</Mapping>
    <Mapping>Novexx</Mapping>
  </AveryNovexx>
  <TextReplacement>
    <Mapping>Avery YT100</Mapping>
    <Mapping>Novexx YT104</Mapping>
    <Mapping>Zebra 0</Mapping>
  </TextReplacement>
</FontMappings>
```

## 9.7. Automatización de informes

EDITION AVAILABILITY:	Automation Easy	Automation Pro	Automation Enterprise
			

Combine los datos de su sistema empresarial con etiquetas que contengan **informes**. Cree informes en NiceLabel Designer y complételos con datos de sus sistemas empresariales en NiceLabel Automation.

Los informes completados utilizan los datos de su sistema empresarial para la impresión. NiceLabel Automation

- **recibe** los **datos** de su sistema empresarial.
- **analiza los datos** con un **filtro**.
- **completa el informe** con datos analizados.
- **ejecuta la impresión** de los nuevos informes completados con un **desencadenador**.

## 9.7.1. Creación de bases de datos temporales

Debe diseñar los **informes** antes de automatizarlos. Para hacerlo, necesita conectarlos a una **base de datos**.

Si utiliza datos de sistemas empresariales externos para los informes, entonces **no tendrá una base de datos** para utilizar. Para funcionar de forma correcta, los datos de informes necesitan una **estructura jerárquica** con elementos definidos claramente.

Cree una **base de datos temporal** con los datos de su sistema empresarial (por lo general, datos de XML o JSON). Una vez que tenga la base de datos temporal, diseñe y configure los informes. **Las bases de datos temporales son solo para la configuración**. Los informes configurados imprimen utilizando desencadenadores de automatización, no las bases de datos temporales.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <!DOCTYPE label SYSTEM "label.dtd">
3 <label FORMAT="/MS1/ERL_FBB/APOLLO_KOMPLISTE.tbl" JOBNAME="SAP" QUANTITY="1" PRINTERNAME="PDF">
4 <label>
5   <variable name="#AVAYOUT"/><MS1/ERL_FBB/APOLLO_KOMPLISTE/><variable>
6   <variable name="#LOCATION"/><ERL_FBB/><variable>
7   <variable name="#SYSTEM"/><SAP/><variable>
8   <variable name="#IDENTICAL_COPIES"/><01/><variable>
9   <variable name="#OFFSETX"/><0/><variable>
10  <variable name="#OFFSETY"/><0/><variable>
11  <variable name="#SKIP"/><00/><variable>
12  <variable name="#MAGNETISS"/><variable>
13  <variable name="#REICH_01"/><AKL/><variable>
14  <variable name="Q_ZZ_01"/><1301/><variable>
15  <variable name="Q_VV_01"/><03/><variable>
16  <variable name="Q_TT_01"/><01/><variable>
17  <variable name="ZUPPOZ_W_01"/><000007/><variable>
18  <variable name="BHT_W_01"/><349665/><variable>
19  <variable name="LGE_W_01"/><001/><variable>
20  <variable name="SME_D1"/><AS0803090/><variable>
21  <variable name="MLF_01"/><6FC532-0A030-0AA1/><variable>
22  <variable name="RENEW_01"/><NCU20.3 PN 6FC532-0A030-0AA1/><variable>
23  <variable name="REWE_S01_01"/><1/><variable>
24  <variable name="GMS_BHT_01"/><0/><variable>
25  <variable name="#REICH_02"/><AKL/><variable>
26  <variable name="Q_ZZ_02"/><1301/><variable>
27  <variable name="Q_VV_02"/><07/><variable>
28  <variable name="Q_TT_02"/><07/><variable>
29  <variable name="ZUPPOZ_W_02"/><000010/><variable>
30  <variable name="BHT_W_02"/><339578/><variable>
31  <variable name="LGE_W_02"/><001/><variable>
32  <variable name="SME_D2"/><A04901000/><variable>
33  <variable name="MLF_02"/><6FC519-00F52-3A00/><variable>
34  <variable name="RENEW_02"/><PCU50.5-C MIN7 6FC519-00F52-3A00/><variable>
35  <variable name="REWE_S01_02"/><1/><variable>
36  <variable name="REWE_S01_03"/><1/><variable>
37  <variable name="REWE_S01_04"/><1/><variable>

```

Datos de sistemas empresariales sin jerarquía.

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <asx:abap xmlns:asx="http://www.sap.com/abapxml" version="1.0">
3   <asx:values>
4     <NICELABEL_JOB>
5       <TIMESTAMP>20130221100527.788134</TIMESTAMP>
6       <USER>PGRI</USER>
7       <IT_LABEL_DATA>
8         <item>
9           <LBL_NAME>goods_receipt.nbl</LBL_NAME>
10          <LBL_PRINTER>Production01</LBL_PRINTER>
11          <LBL_QUANTITY>1</LBL_QUANTITY>
12          <MAKTX>MASS ONE</MAKTX>
13          <MATNR>28346</MATNR>
14          <MEINS>KG</MEINS>
15          <MDATU>19.01.2012</MDATU>
16          <QUANTITY>1</QUANTITY>
17          <EXIDV>012345678901234560</EXIDV>
18        </item>
19        <item>
20          <LBL_NAME>goods_receipt.nbl</LBL_NAME>
21          <LBL_PRINTER>Production01</LBL_PRINTER>
22          <LBL_QUANTITY>1</LBL_QUANTITY>
23          <MAKTX>MASS TWO</MAKTX>
24          <MATNR>28346</MATNR>
25          <MEINS>KG</MEINS>
26          <MDATU>11.01.2011</MDATU>
27          <QUANTITY>1</QUANTITY>
28          <EXIDV>012345678901234577</EXIDV>
29        </item>
30        <item>
31          <LBL_NAME>goods_receipt.nbl</LBL_NAME>
32          <LBL_PRINTER>Production01</LBL_PRINTER>
33          <LBL_QUANTITY>1</LBL_QUANTITY>
34          <MAKTX>MASS THREE</MAKTX>
35          <MATNR>27844</MATNR>
36          <MEINS>KG</MEINS>
37          <MDATU>07.03.2009</MDATU>
38          <QUANTITY>1</QUANTITY>
39          <EXIDV>012345678901234584</EXIDV>
40        </item>
41      </IT_LABEL_DATA>
42    </NICELABEL_JOB>
43  </asx:values>
44 </asx:abap>

```

Datos con jerarquía que pueden utilizarse para imprimir informes.

```

<?xml version="1.0" encoding="UTF-8"?>
<dataroot xmlns:od="urn:schemas-microsoft-com:officedata" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="Pasta.xsd" generated="2020-08-31T15:43:51">
  <Total>211</Total>
  <Pasta>
    <ProductID>PAS501</ProductID>
    <CodeEAN>8021228310001</CodeEAN>
    <ProductName>BIGOLI 250G</ProductName>
    <Package>6</Package>
    <category>Long pasta</category>
    <Description>A long, thin, cylindrical pasta of Italian origin. Spaghetti is made of semolina or flour and water.</Description>
  </Pasta>
  <Pasta>
    <ProductID>PAS502GI</ProductID>
    <CodeEAN>8021228310018</CodeEAN>
    <ProductName>TAGLIATELLE 250G</ProductName>
    <Package>6</Package>
    <category>Ribbon-cut pasta</category>
    <Description>Ribbon style pasta are often rolled flat and then cut. This can be done by hand or mechanically.</Description>
  </Pasta>

```

Ejemplo de datos XML que necesita para diseñar informes (sin analizar).

Para las bases de datos temporales, cree un archivo de base de datos de texto en formato CSV. Tiene varias opciones:

1. Convierta la estructura de los datos a un archivo de texto CVS de forma manual:



*Cree bases de datos de texto temporales en formato CSV de forma manual.*

2. Utilice una herramienta de conversión común de XML a Excel o de XML a CSV y organice los datos de forma manual en una hoja de cálculo.
3. Utilice un **filtro de datos** de automatización para convertir los datos de forma automática en una base de datos de texto en formato CSV (consulte el archivo de muestra de **XML a CSV** para obtener más información).

## 9.7.2. Diseño de informes automatizados

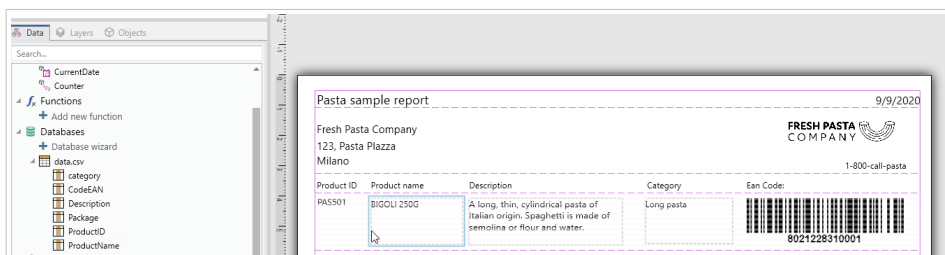
Abra **Designer** para crear un **informe** con la base de datos temporal en formato CVS:

1. **Conecte** la base de datos CSV temporal al **informe**.
2. **Diseñe** el informe utilizando los datos de texto del archivo CSV como variables para los objetos en las **definiciones de reiteración**.



### NOTA

Asegúrese de que los **nombres de las variables** coincidan con los **nombres de los datos** en el filtro para asignar los datos de forma correcta.

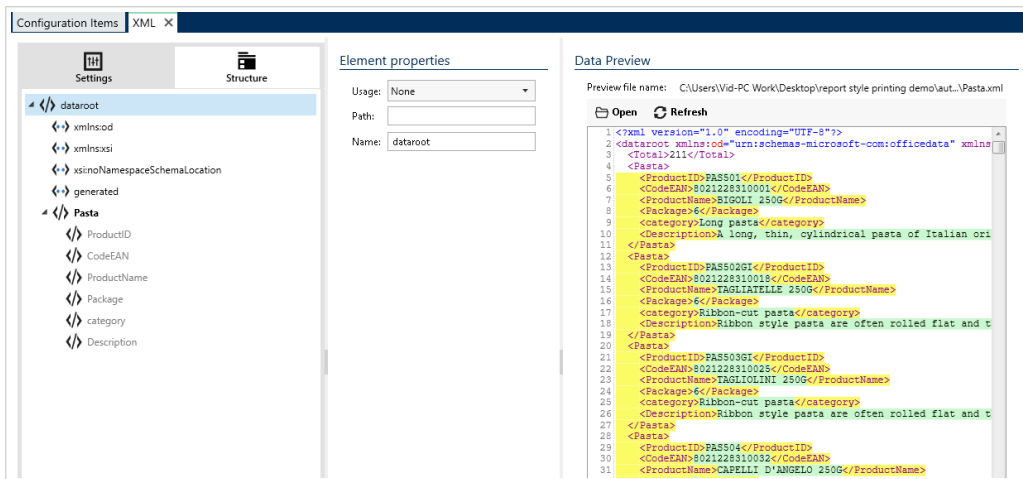


*Diseño de informes con base de datos temporales en formato CSV.*

## 9.7.3. Creación de filtros de datos

Cree un **filtro de datos** en Automatización para usar en los informes (consulte el desencadenador de muestra para obtener más información sobre la creación de filtros XML):





Configuración del filtro de datos XML.

Consejos para crear filtros para informes:

- Utilice el **Asistente de importación de estructura** para importar la estructura de datos.
- Nombre los elementos en bloques de datos de elementos reiterativos.
- Configure los elementos como áreas de asignación o los subelementos como variables.

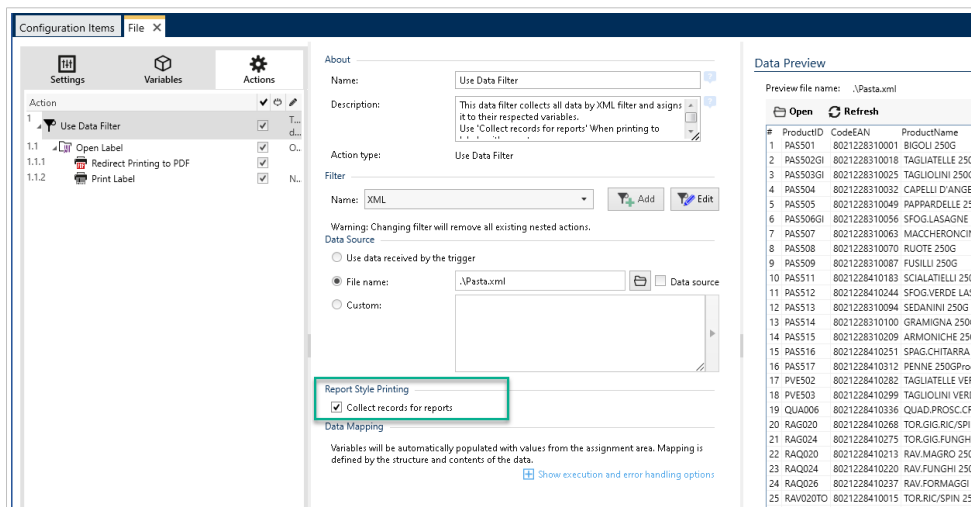
## 9.7.4. Creación de desencadenadores para los nuevos filtros de datos



### NOTA

Para que el desencadenador funcione, los nombres de las fuentes de datos en Designer tienen que coincidir con los nombres de los bloques de datos en el filtro.

- Incluya la acción **Usar filtro de datos** para aplicar el filtro.



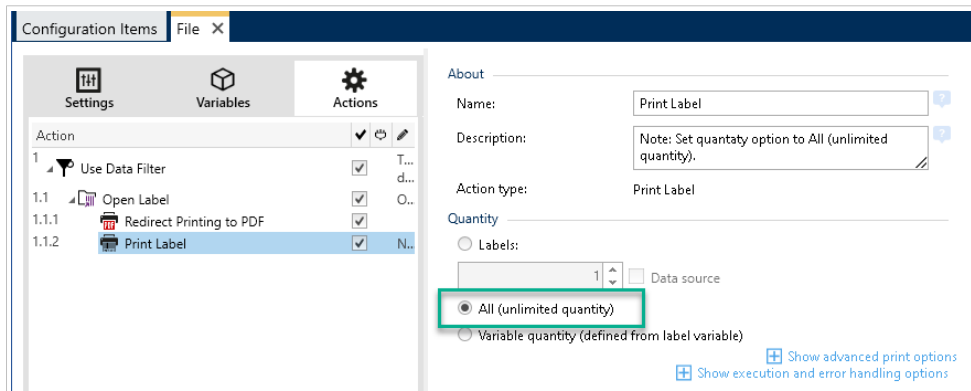
- Habilite la opción **Recopilar registros para informes** para recopilar todos los registros en una tabla que Automatización utilizará para abrir las etiquetas de informes.



## NOTA

Utilice los mismos nombres de estructuras de datos en los informes, desencadenadores y filtros.

- Incluya la acción **Imprimir etiqueta**.



- Habilite la opción **Todas (cantidad ilimitada)**.

Ejecute el desencadenador. Ahora el informe se actualizará de forma automática con nuevos datos e imprimirá con el desencadenador.

## 9.8. Cambiar configuración predeterminada de impresión multiproceso

EDITION AVAILABILITY:

Automation Easy



Automation Pro



Automation Enterprise



Cada NiceLabel Automation producto puede aprovechar múltiples núcleos de procesamiento - cada núcleo ejecuta un proceso de impresión en forma independiente. La mitad de los núcleos se utilizan para realizar subprocesos *normales* simultáneos, y la otra mitad, para ejecutar subprocesos *de impresión de sesiones simultáneas*.



## NOTA

En circunstancias normales, nunca tendrá que cambiar la configuración predeterminada. Asegúrese de tener la información correcta al cambiar esa configuración.

Para modificar el número de subprocesos de impresión simultáneos, realice lo siguiente:

1. Abra el archivo `product.config` en el editor de texto.  
El archivo se encuentra aquí:

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10\product.config
```

2. Cambie los valores de los elementos **ProcesosDelImpresiónSimultáneosMáx** y **ProcesosDelImpresiónDeSesionesSimultáneosMáx**.

```
<configuration>
  <IntegrationService>
    <MaxConcurrentPrintProcesses>1</MaxConcurrentPrintProcesses>
    <MaxConcurrentSessionPrintProcesses>1</
MaxConcurrentSessionPrintProcesses>
  </IntegrationService>
</configuration>
```

3. Guarde el archivo. NiceLabel Automation actualizará automáticamente el servicio con el nuevo número de subprocesos de impresión.

## 9.9. Compatibilidad con productos NiceWatch

NiceLabel Automation le permite cargar las configuraciones de desencadenadores que se crearon usando productos NiceWatch heredados. En la mayoría de los casos, puede ejecutar la configuración de NiceWatch en NiceLabel Automation sin ninguna modificación.

Los productos NiceLabel Automation utilizan un nuevo motor de impresión basado en .NET optimizado para un mejor rendimiento y una menor superficie de memoria. El nuevo motor de impresión no admite cada opción de diseño de etiqueta que está disponible en el diseñador de etiquetas. Cada versión nueva de NiceLabel Automation reduce las diferencias, pero es posible que se encuentre con que algunas funciones no están disponibles.

## Resolver problemas de incompatibilidad

NiceLabel Automation le advertirá si intenta imprimir plantillas de etiquetas existentes que contengan la funcionalidad de diseño, que no está disponible en el nuevo motor de impresión.

Si hay incompatibilidades con las plantillas de etiquetas o archivos de configuración de NiceWatch, Automation será notificado acerca de:

- **Compatibilidad con la configuración del desencadenador:** Al abrir la configuración de NiceWatch (archivo .MIS), NiceLabel Automation lo comprueba en comparación con las funciones compatibles. No todas las funciones de los productos NiceWatch están disponibles en NiceLabel Automation. Si bien algunas no están disponibles, otras están configuradas de manera diferente. Si el archivo MIS contiene funciones no compatibles, las verá en una lista. Automation elimina estas funciones de la configuración.  
En este caso, debe abrir el archivo .MIS en Automation Builder y resolver los problemas de incompatibilidad. Deberá usar las funciones NiceLabel Automation disponibles para recrear los elementos de la configuración eliminada.
- **Compatibilidad con plantillas de etiquetas:** Si sus plantillas de etiquetas existentes contienen funciones de motor de impresión no compatibles, tal lo proporcionado por NiceLabel Automation, verá mensajes de error en el panel **Registro** . Esta información se ve en Automation Builder (al diseñar desencadenadores) o en Automation Manager (al ejecutar los desencadenadores).  
En este caso, debe abrir el archivo de la etiqueta en el diseñador de etiquetas y eliminar las funciones no admitidas de la etiqueta.



### NOTA

Para obtener más información acerca de los problemas de incompatibilidad con NiceWatch y los diseñadores de etiquetas, consulte [NiceLabel Automation Compatibility with NiceLabel Designers V6 and NiceWatch V5](#).

## Abrir la configuración de NiceWatch para editar

Abra la configuración de NiceWatch existente (archivo .MIS) en Automation Builder y edítela en Automation Builder. Puede guardar la configuración como archivo .MISX solamente.

Para editar la configuración de NiceWatch, realice lo siguiente:

1. Abra Automation Builder.
2. Vaya a **Archivo > Abrir archivo de NiceWatch**.
3. En el cuadro de diálogo **Abrir** , busque el archivo de configuración de NiceWatch (archivo .MIS).
4. Haga clic en **Aceptar**.
5. Si la configuración contiene funciones no compatibles, estas se muestran en una lista. Automation las elimina de la configuración.

### Abrir la configuración de NiceWatch para ejecución

Puede abrir la configuración NiceWatch (archivo .MIS) en Automation Manager sin conversión al formato de archivo NiceLabel Automation (archivo .MISX). Si los desencadenadores de NiceWatch son compatibles con NiceLabel Automation, puede comenzar a usarlos enseguida.

Para editar e implementar la configuración de NiceWatch, realice lo siguiente:

1. Abra Automation Manager.
2. Haga clic en el botón **Agregar**.
3. En el cuadro de diálogo **Abrir**, cambie el tipo de archivo por **Configuración de NiceWatch**.
4. Busque el archivo de configuración de NiceWatch (archivo .MIS).
5. Haga clic en **Aceptar**.
6. Automation Manager mostrará el desencadenador de la configuración seleccionada. Para iniciar el desencadenador, selecciónelo y haga clic en el botón **Iniciar**.



#### NOTA

Si hay algún problema de compatibilidad con la configuración de NiceWatch, deberá abrirla en Automation Builder y reconfigurarla.

## 9.10. Control del servicio Automation con parámetros de línea de comandos

Lea esta sección para saber cómo usar el símbolo del sistema para:

- Iniciar o detener servicios Automation.
- Controlar qué configuraciones se cargan.
- Controlar qué desencadenadores están activos.



#### NOTA

Asegúrese de ejecutar el **Símbolo del sistema** en el modo elevado (con permisos administrativos). Haga clic con el botón derecho en `cmd.exe` y seleccione **Ejecutar como administrador**.

## Inicio y detención de servicios

Para iniciar ambos servicios, utilice los siguientes comandos:

```
net start NiceLabelProxyService 10  
net start NiceLabelAutomationService 10
```

Si quiere abrir el archivo de configuración una vez iniciado el Servicio, utilice:

```
net start NiceLabelAutomationService 10 [Configuration]
```

Por ejemplo:

```
net start NiceLabelAutomationService 10 "c:\Project\configuration.MISX"
```

Para detener servicios, utilice los siguientes comandos:

```
net stop NiceLabelProxyService 10  
net stop NiceLabelAutomationService 10
```

## Administración de configuraciones y desencadenadores

El servicio NiceLabel Automation puede controlarse con los parámetros de línea de comandos Automation Manager. Use la siguiente sintaxis general:

```
NiceLabelAutomationManager.exe Configuración de COMANDOS [TriggerName:] [/SHOWUI]
```



### NOTA

Incluya la ruta completa al nombre de la configuración. No use el nombre del archivo solamente.

## Para AGREGAR configuración

La configuración provista se cargará en el servicio. No se inicia ningún desencadenador. Si incluye el parámetro `/SHOWUI`, Automation Manager se inicia la IU.

```
NiceLabelAutomationManager.exe AGREGUE c:\Project\configuration.MISX /SHOWUI
```

## Para VOLVER A CARGAR configuración

La configuración provista se cargará en el servicio. Se preservará el estado de ejecución de todos los desencadenadores. Si se vuelve a cargar la configuración, se fuerza la actualización de todos los archivos almacenados en caché para esta configuración. Para obtener más información, consulte el tema [Almacenar archivos en caché](#). Si incluye el parámetro `/SHOWUI`, Automation Manager se inicia la IU.

```
NiceLabelAutomationManager.exe RELOAD c:\Project\configuration.MISX /SHOWUI
```

## Para ELIMINAR configuración

Se descargarán del servicio la configuración provista y todos los desencadenadores.

```
NiceLabelAutomationManager.exe REMOVE c:\Project\configuration.MISX
```

## Para INICIAR un desencadenador

El desencadenador de referencia se iniciará en la configuración que ya está cargada.

```
NiceLabelAutomationManager.exe START c:\Project\configuration.MISX CSVTrigger
```

## Para DETENER un desencadenador

El desencadenador de referencia se detendrá en la configuración que ya está cargada.

```
NiceLabelAutomationManager.exe DETENGA c:\Project\configuration.MISX CSVTrigger
```

### Códigos de estado

Los códigos de estado proporcionan comentarios sobre la ejecución de líneas de comandos. Para activar el retorno de los códigos de estado, use la siguiente sintaxis de línea de comandos:

```
iniciar / esperarNiceLabelAutomationManager.exe Configuración de COMANDO  
[TriggerName] [/SHOWUI]
```

Los códigos de estado se capturan en la variable del sistema `errorlevel`. Para ver el código de estado, ejecute el siguiente comando:

```
echo %errorlevel%
```

Lista de códigos de estado:

Código de estado	Descripción
0	No ocurrieron errores
100	No se encontró el nombre de archivo de configuración
101	La configuración no se puede cargar
200	No se encontró el desencadenador
201	No puede iniciar el desencadenador

### Proporcionar credenciales de usuario para la Autenticación de aplicaciones

Si ha configurado LMS Enterprise o el sistema LMS Pro para utilizar la **Autenticación de aplicaciones** (no la **Autenticación de Windows**), debe proporcionar credenciales de usuario con suficientes permisos para administrar las configuraciones y los desencadenadores.

Hay dos parámetros de línea de comandos que puede utilizar:

- `-USER:[username]`. Donde `[username]` es un marcador de posición para el nombre de usuario real.
- `-PASSWORD:[password]`. Donde `[password]` es un marcador de posición para la contraseña real.



## 9.11. Reemplazo de cadena de conexión de bases de datos

Un archivo de configuración para el Servicio de Automation puede incluir comandos de reemplazo de cadenas de conexión de bases de datos.

El usuario puede configurar el servicio para reemplazar ciertas partes de la cadena de conexión mientras se ejecuta el desencadenador. Esto permite que una instancia única de Automation use la misma configuración, pero en realidad utiliza un servidor de bases de datos diferente para la funcionalidad relacionada a la base de datos. Esto permite al usuario configurar desencadenadores en los entornos de desarrollo y ejecutarlos en los entornos de producción sin ningún cambio en la configuración.

La lógica de reemplazo de la cadena de conexión está definida en el archivo llamado `DatabaseConnections.Config` en la carpeta del sistema de Automation.

1. Abra la carpeta Sistema de Automation `%PROGRAMDATA%\NiceLabel\NiceLabel 10`
2. Cree un archivo con el nombre `DatabaseConnections.config`.
3. Abra el archivo `DatabaseConnections.config` recién creado en un editor de texto e introduzca el contenido de muestra:

```
<?xml version="1.0" encoding="UTF-8"?>
<DatabaseConnectionReplacements>
  <Replacement>
    <From>Data Source=mysqlServer</From>
    <To>Data Source=NEW_mysqlServer</To>
  </Replacement>
  <Replacement>
    <From>Initial Catalog=myDatabase</From>
    <To>Initial Catalog=NEW_myDatabase</To>
  </Replacement>
</DatabaseConnectionReplacements>
```

Este XML de muestra es un ejemplo, donde un desencadenador existente tiene una conexión con el servidor de Microsoft SQL llamado `mysqlServer` y la base de datos llamada `myDatabase`. Debe actualizar la cadena de conexión para usar la base de datos `NEW_myDatabase` en el servidor `NEW_mysqlServer`.

Se deben definir dos elementos de Reemplazo, uno para cambiar el nombre del servidor y uno para cambiar el nombre de la base de datos.

4. Verifique la cadena de conexión existente en sus etiquetas y, luego, cambie el contenido de `<Replacement>` usando los valores correctos.

El archivo de configuración define los pares desde-hasta en la estructura XML. La etiqueta `<Replacement>` contiene un elemento `<From>` y un elemento `<To>`. Durante la ejecución de desencadenadores, la cadena "desde" se reemplaza con la cadena "hasta". Puede definir todos los elementos `<Replacement>` que sean necesarios.



## NOTA

Algunas cadenas de conexión utilizan Server=servername, mientras que otras utilizan Data Source=servername. Si utiliza la misma base de datos, no es necesario introducir nombres de bases de datos.

5. Edite su archivo `product.config` que se encuentra en `%PROGRAMDATA%\NiceLabel\NiceLabel 10`. Añada el elemento `/Common/General/UseLocalReplacementStrings` y asigne el valor **Verdadero**:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <Common>
    <General>
      ...
      <UseLocalReplacementStrings>True</UseLocalReplacementStrings>
    </General>
  </Common>
  ...
</configuration>
```

6. Reinicie ambos servicios de Automation después de haber agregado o editado los archivos de configuración.
7. Repita los pasos 1 a 6 en todas las computadoras que ejecuten las configuraciones de Automation donde desee reemplazar las cadenas de conexión de las bases de datos.



## NOTA

El reemplazo de cadenas de conexión de las bases de datos no modifica las cadenas de conexión de sus archivos de etiquetas. Este proceso simplemente anula las cadenas de conexión cuando imprime desde Automation.

## 9.12. Introducción de caracteres especiales (Códigos de control)

Los caracteres especiales o códigos de control son caracteres binarios que no están representados en el teclado. No puede escribirlos como se escriben los caracteres normales porque deben estar codificados usando una sintaxis especial. Necesitaría usar dichos caracteres cuando se comunique con dispositivos de puertos de serie, cuando reciba datos en el puerto TCP/IP, o cuando trabaje con archivos binarios, como archivos de impresión.

Hay dos métodos para introducir caracteres especiales:

- **Ingresar los caracteres manualmente** utilizando uno de los ejemplos de sintaxis descritos:

- Use la sintaxis `<special_character_acronim>`, por ejemplo `<FF>` para Avance de página, o `<CR>` para Retorno de carro, o `<CR><LF>` para una línea nueva.
- Use la sintaxis `<#hex_code>`, como `<#0D>` (decimal 13) para Retorno de carro o `<#00>` para carácter nulo.

Para obtener más información, consulte el tema [Lista de códigos de control](#).

- **Insertar los caracteres de la lista.** Los objetos que admiten caracteres especiales como contenido tienen un botón de flecha en el lado derecho. El botón tiene un acceso directo para todos los caracteres especiales disponibles. Cuando seleccione un carácter de la lista, se agrega al contenido. Para obtener más información, consulte la sección [Usar valores compuestos](#).

## 9.13. Lista de códigos de control

Código ASCII	Abreviatura	Descripción
1	SOH	Principio de encabezado
2	STX	Principio de texto
3	ETX	Final de texto
4	EOT	Fin de la transmisión
5	ENQ	Consulta
6	ACK	Confirmación
7	BEL	Campana
8	BS	Retroceso
9	HT	Pestaña horizontal
10	LF	Avance de línea
11	VT	Pestaña vertical
12	FF	Avance de página
13	CR	Retorno de carro
14	SO	Desplazar hacia afuera
15	SI	Desplazar hacia adentro
16	DLE	Escape de vínculo de datos
17	DC1	XON - Control de dispositivo 1
18	DC2	Control de dispositivo 2
19	DC3	XOFF - Control de dispositivo 3
20	DC4	Control de dispositivo 4
21	NAK	Confirmación negativa
22	SYN	Inactividad sincrónica
23	ETB	Fin de la transmisión del bloque

24	CAN	Cancelar
25	EM	Fin del medio
26	SUB	Sustituir
27	ESC	Escape
28	FS	Separador de archivos
29	GS	Separador de grupos
30	RS	Separador de registros
31	US	Separador de unidades
188	FNC1	Código de funciones 1
189	FNC2	Código de funciones 2
190	FNC3	Código de funciones 3
191	FNC4	Código de funciones 4

## 9.14. Licencias y uso de impresora

En función del tipo de licencia, su copia del producto NiceLabel puede estar limitado al uso simultáneo de un número limitado de impresoras. En caso de una licencia para múltiples usuarios, NiceLabel mantiene un registro del número y los nombres de las diferentes impresoras que se usan para imprimir en todos los clientes NiceLabel de su entorno. El identificador de impresora único es una combinación del nombre del controlador de la impresora (no el nombre de la impresora) y de la ubicación y el puerto de la impresora.

"Para usar una impresora" significa que una de las acciones detalladas a continuación se ha realizado en la configuración de Automation:

- [Imprimir etiqueta](#)
- [Enviar datos a impresora](#)
- [Generar vista previa de etiqueta](#)
- [Establecer parámetro de impresión](#)

Cada una de estas acciones indica que se ha usado una impresora. La impresora asociada se agrega a la lista de impresoras usadas y permanece en la lista por 7 días desde el último uso. Para eliminar una impresora de la lista, no la utilice por un período de 7 días y se eliminará automáticamente. El software mostrará la información de **Último uso** para que sepa cuándo pasarán los 7 días para cada impresora. Para enlazar un puesto de impresora con una impresora específica, haga clic en la casilla de verificación **Reservada**. La reserva asegura que la impresora esté disponible en todo momento; aun si una impresora ha estado inactiva por más de 7 días.



## AVISO

Si supera el número de puestos definidos por su licencia, el software ingresa en un modo de gracia de 30 días. Mientras se encuentre en este modo, el número de impresoras permitidas aumentará temporalmente al doble de las impresoras permitidas por licencia.

El período de gracia brinda una cantidad de tiempo suficiente para resolver problemas de licencia sin inactividad en la impresión ni pérdida de la capacidad de diseñar etiquetas. El número en exceso de impresoras permitidas generalmente es un efecto del reemplazo de impresoras en su entorno. Sucede cuando se usan impresoras antiguas y nuevas simultáneamente, o si agrega nuevas impresoras. Si no resuelve la violación de la licencia dentro del período de gracia de 30 días, el número de impresoras disponibles disminuirá al número de puestos comprados, comenzando por las impresoras de la lista que se utilizaron por última vez.



## SUGERENCIA

Para conocer más acerca de la NiceLabel 10 licencias, [lea el documento exclusivo](#) - NiceLabel 10 Licencias.

## 9.15. Ejecutar en modo de servicio

NiceLabel Automation se ejecuta como un servicio de Windows y está diseñado de modo que no requiera ninguna intervención del usuario cuando procesa datos y ejecuta acciones. El servicio está configurado para abrirse cuando arranque el sistema operativo y se ejecuta en segundo plano siempre que se esté ejecutando Windows. NiceLabel Automation recordará la lista de todas las configuraciones cargadas y los desencadenadores activos. Cuando se reinicia el servidor, se restablece el último estado conocido.

El servicio se ejecuta con los privilegios de la cuenta de usuario seleccionada durante la instalación. El servicio heredará todos los permisos de acceso de esa cuenta de usuario, incluido el acceso a los recursos de red compartidos, como los controladores de redes e impresoras. Use la cuenta de algún usuario existente con privilegios suficientes, o mejor aún, cree una cuenta dedicada solo para NiceLabel Automation.

Puede administrar el servicio abriendo los Servicios del panel de control de Windows. En sistemas operativos de Windows modernos, también se puede administrar el servicio en la pestaña Servicios del Administrador de tareas de Windows. Utilizará Servicios para ejecutar algunas tareas como:

- Inicio y detención de servicios.
- Cambio de cuenta con la cual se registra el servicio.

## Buenas prácticas de configuración de la cuenta de usuario para el Servicio

- Aunque es posible, se considera una práctica incorrecta ejecutar el servicio con la Cuenta local del sistema. Es una cuenta local de Windows predefinida con privilegios extendidos en la computadora local, pero generalmente no tiene privilegios para acceder a los recursos de red. NiceLabel Automation necesita acceso total a la carpeta %temp% de la cuenta, que puede no estar disponible para la Cuenta local del sistema.
- Si crea una **cuenta de usuario nueva** para el servicio NiceLabel Automation, asegúrese de registrarse en Windows con este nuevo usuario al menos una vez. De este modo, se garantiza que la cuenta de usuario está creada por completo. Por ejemplo, cuando inicia sesión, se crea la carpeta temporal %temp%.
- Deshabilite el requisito para cambiar la contraseña ocasionalmente para esta cuenta de usuario.
- Asegúrese de que la cuenta tenga permisos para **Iniciar sesión como servicio**.
- Ejecute el Servicio en modo x64 (64 bits).

## Acceso a recursos

NiceLabel Automation hereda todos los privilegios de la cuenta de usuario de Windows con la que se ejecuta el servicio. El servicio ejecuta todas las acciones con el nombre de esa cuenta. Se puede abrir la etiqueta si la cuenta tiene permisos para acceder al archivo. Se puede imprimir la etiqueta si la cuenta tiene acceso al controlador de la impresora.

Al usar el sistema de control de revisión y los pasos de aprobación dentro del Almacenamiento de documentos en el Centro de control, tiene que hacer que el miembro de la cuenta de usuario del servicio tenga el perfil de "solo impresión", como el **Operador**. Cuando termine, configure los permisos de acceso para la carpeta específica en modo de **solo lectura** o perfil de Operador. De este modo, se asegura de que NiceLabel Automation solamente usará las etiquetas aprobadas, no borradores.

Para obtener más información, consulte el tema [Acceso a recursos de red compartidos](#).

### Modo de servicio: 32 bits o 64 bits

NiceLabel Automation puede ejecutarse en sistemas de 32 bits (x86) y 64 bits (x64) en forma nativa.

El modo de ejecución lo determina automáticamente el sistema operativo de Windows. NiceLabel Automation se ejecutará en el modo de 64 bits en Windows de 64 bits y en el modo de 32 bits en Windows de 32 bits.

- **Impresión:** Si se ejecuta Automation como proceso de 64 bits existen algunos beneficios, tales como la comunicación directa con el Administrador de trabajos de impresión de la impresora de 64 bits en Windows de 64 bits. Así se eliminan los conocidos problemas con el SPLWOW64.EXE, un software intermedio que habilita a aplicaciones de 32 bits a usar el servicio del administrador de trabajos de impresión de la impresora de 64 bits.
- **Acceso a base de datos:** La ejecución como proceso de 64 bits del Servicio NiceLabel Automation requiere la versión de 64 bits de los controladores de la base de datos para poder acceder a los datos. Para obtener más información, consulte el tema [Acceso a bases de datos](#).



#### NOTA

Si no tiene controladores de bases de datos de 64 bits para su base de datos, no puede usar NiceLabel Automation en modo de 64 bits. Tiene que instalarlo en un sistema de 32 bits, o forzarlo a ejecutarse en el modo de 32 bits.

### Forzar el modo de operación x86 en Windows x64

Puede haber motivos para ejecutar NiceLabel Automation como aplicación de 32 bits en Windows de 64 bits.

Para forzar NiceLabel Automation en el modo x86 en Windows x64, realice lo siguiente:

- Seleccione Inicio > Ejecutar.
- Escriba **regedit** y presione Enter.
- Navegue a la tecla

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\NiceLabelAutomationService
```

- Cambie el nombre del archivo por NiceLabelAutomationService10.x86.exe, pero mantenga la ruta de acceso existente.
- Reinicie el servicio NiceLabel Automation.



#### AVISO

No se recomienda cambiar el modo de servicio NiceLabel Automation. Si decide cambiarlo de todas formas, asegúrese de que realice una prueba exhaustiva del desencadenador antes de implementar la configuración en entorno de producción.

## 9.16. Orden de búsqueda de los archivos solicitados

Cuando intente cargar un archivo de imagen o archivo de etiqueta específico, NiceLabel Automation lo buscará en las diversas ubicaciones.

NiceLabel Automation ubica el archivo en el siguiente orden:

1. Verifica si el archivo existe en la ubicación especificada por la acción.
2. Verifica si el archivo existe en la misma carpeta que el archivo de configuración (.MISX).
3. Verifica si el archivo existe en la carpeta .\Etiquetas (para archivos de gráficos, verifica la carpeta .\Gráficos).
4. Verifica si el archivo existe en la carpeta ..\Etiquetas (para archivos de gráficos, verifica la carpeta ..\Gráficos).
5. Verifica si el archivo existe en la carpeta global Etiquetas (carpeta Gráficos para archivos de gráficos), según la configuración de las opciones.

Si no encuentra el archivo en ninguna de esas ubicaciones, falla la acción y genera el error.



## 9.17. Asegurar el acceso a sus desencadenadores

Ciertas implementaciones requieren acceso seguro a los desencadenadores. NiceLabel Automation le permite activar medidas de seguridad para otorgar acceso de los desencadenadores solo a dispositivos de red confiables. La configuración de seguridad depende del tipo de desencadenador. Algunos tipos de desencadenadores permiten configurar el acceso a la seguridad por diseño. Para todos los desencadenadores basados en el protocolo TCP/IP, también puede definir todos los detalles dentro de firewall de Windows.

### Configuración de firewall

Al usar desencadenadores basados en TCP/IP, como [Desencadenador de servidor TCP/IP](#), [Desencadenador de servidor HTTP](#) o [Desencadenador de servicio Web](#) asegúrese de permitir que las aplicaciones externas se conecten a los desencadenadores. Cada desencadenador se ejecuta dentro del servicio NiceLabel Automation, cuyo acceso está regido por el firewall de Windows.



#### NOTA

En forma predeterminada, el firewall de Windows se configura para permitir todas las conexiones entrantes al servicio NiceLabel Automation. Esto facilita la configuración y la prueba de los desencadenadores, pero puede quedar sujeto a accesos no autorizados.

Si la implementación de NiceLabel Automation en su compañía está sujeta a regulaciones de seguridad estrictas, debe actualizar las reglas del firewall de acuerdo con ellas.

Por ejemplo:

- Puede definir que el firewall acepte el tráfico entrante de fuentes conocidas únicamente.
- Puede permitir el ingreso de datos en puertos predefinidos únicamente.
- Puede permitir la conexión de ciertos usuarios únicamente.
- Puede definir en qué interfaces aceptará la conexión entrante.

Para realizar cambios en el firewall de Windows, abra la consola de administración **Firewall de Windows con Seguridad avanzada** en **Panel de control > Sistema y seguridad > Firewall de Windows > Configuración avanzada**.



#### NOTA

Cuando NiceLabel Automation está vinculado a productos NiceLabel Control Center, asegúrese de habilitar la conexión entrante del puerto **56415/TCP**. Si cierra este puerto, no podrá administrar NiceLabel Automation desde el Centro de control.

### **Permitir acceso según los permisos de acceso de archivos**

El desencadenador de archivos se ejecutará luego del evento de cambio-marca-hora en el archivo o los archivos monitoreados. Debe incluir los archivos de desencadenadores en una carpeta, a la cual pueda acceder el servicio NiceLabel Automation. La cuenta de usuario que ejecuta el Servicio debe tener acceso a los archivos. Al mismo tiempo, los permisos de acceso a la ubicación también determinan qué usuario o aplicación puede guardar el archivo de desencadenadores. Debe configurar los permisos de acceso de modo que únicamente los usuarios autorizados puedan guardar archivos.

### **Permitir acceso según la dirección IP y el nombre de host**

Puede proteger el acceso al desencadenador del servidor TCP/IP con dos listas de direcciones IP y nombres de hosts.

- La primera lista "**Permitir conexiones de los siguientes hosts**" contiene las direcciones IP o nombres de hosts de dispositivos que pueden enviar datos al desencadenador. Cuando algún dispositivo tiene sus direcciones IP incluidas en esta lista, puede enviar datos al desencadenador.
- La segunda lista '**No permitir conexiones de los siguientes hosts**' contiene las direcciones IP o nombres de hosts de dispositivos que no pueden enviar datos. Cuando algún dispositivo tiene sus direcciones IP incluidas en esta lista, no puede enviar datos al desencadenador.

### **Permitir acceso según los nombres de usuarios o contraseñas**

Puede proteger el acceso al desencadenador del servidor HTTP activando la autenticación del usuario. Al estar activada, cada solicitud HTTP enviada al desencadenador del servidor HTTP debe incluir la combinación '**nombre de usuario y contraseña**' que coincida con la combinación definida.

### **Permitir acceso según la membresía del grupo de aplicación**

Puede proteger el acceso al desencadenador de servidor HTTP agregando usuarios a un grupo de aplicación en el Centro de control. Con esta opción habilitada, solo los miembros autenticados de este grupo podrán acceder al desencadenador.

## **9.18. Impresión de sesiones**

La impresión de sesiones permite imprimir múltiples etiquetas utilizando un único trabajo de impresión. Si la impresión de sesiones está habilitada, la impresora recibe, procesa e imprime todas las etiquetas del trabajo de impresión de una vez. Como resultado, aumenta la velocidad de impresión debido al proceso continuo de impresión de etiquetas agrupadas.



## SUGERENCIA

La impresión de sesiones sirve de alternativa a la impresión sin sesiones que se utiliza normalmente, durante la cual cada etiqueta se envía a una impresora como un trabajo de impresión separado.



## NOTA

Automation Activa la impresión de sesiones automáticamente según la configuración de acciones.

### ¿Cómo comienza la impresión de sesiones?

La impresión de sesiones comienza automáticamente si las acciones [Para bucle](#), [Para cada registro](#) o [Para cada línea](#) están presentes en el flujo de trabajo. En ese caso, la acción [Imprimir etiqueta](#) habilita automáticamente la impresión de sesiones. Esto significa que las acciones de impresión para todos los elementos del bucle se incluyen en un único trabajo de impresión.

The screenshot shows the 'Automation' interface with three tabs: 'Settings', 'Variables', and 'Actions'. The 'Actions' tab is active, displaying a tree view of actions. The 'Action' list includes:

- 1 For loop
  - 1.1 Open Label
    - 1.1.1 Set Printer
    - 1.1.2 Set Print Job Name
    - 1.1.3 Redirect Printing to File
    - 1.1.4 Print Label

A green callout box with the text 'Session Printing ON' points to the 'For loop' action. To the right of the 'Print Label' action, there are three checked checkboxes.

## ¿Cómo finaliza la impresión de sesiones?

Cada impresión de sesiones finaliza con un bucle finalizado o con una acción [Imprimir etiqueta](#) combinada con al menos una de las siguientes condiciones:

- Cambia la impresora. Si selecciona otra impresora utilizando la acción [Establecer impresora](#), finaliza la impresión de sesiones.
- Cambia el puerto de la impresora. Si redirecciona el trabajo de impresión a un archivo utilizando la acción [redireccionar impresión a archivo](#), finaliza la impresión de sesiones.
- Cambia la etiqueta. Si selecciona otra etiqueta para imprimir utilizando la acción [Abrir etiqueta](#), finaliza la impresión de sesiones.
- Se envía un comando personalizado que finaliza la impresión de sesiones. Si envía el comando `SESSIONEND` utilizando la acción [Enviar comandos personalizados](#), finaliza la impresión de sesiones.



### NOTA

En este caso, `SESSIONEND` debe enviarse como el único elemento en la acción [Enviar comandos personalizados](#). Si desea enviar comandos adicionales, utilice acciones [Enviar comandos personalizados](#) separadas.



### NOTA

Las configuraciones más complejas pueden tener múltiples bucles anidados uno dentro de otro. En ese caso, la impresión de sesiones finaliza cuando se sale del bucle principal de más afuera.

## 9.19. Sugerencias y trucos para usar variables en acciones

Si usa variables en NiceLabel Automation acciones, siga estas recomendaciones.

- **Encierre las variables entre paréntesis.** Cuando haya variables con espacios en el nombre y se refieran a variables de acciones, como [Ejecutar instrucción SQL](#) o [Ejecutar secuencia de comandos](#), encierre las variables entre paréntesis, como `(Nombre del producto)`. También usará paréntesis si los nombres de las variables son los mismos que los nombres reservados, por ejemplo, en la instrucción SQL.
- **Coloque dos puntos frente al nombre de la variable.** Para hacer referencia a la variable en la acción [Ejecutar instrucción SQL](#) o en un [Desencadenador de base de datos](#), debe incluir dos puntos (`:`) frente al nombre de la variable, como `:(Product ID)`. El analizador SQL lo entenderá como 'valor de variable'.

```
SELECT * FROM MyTable WHERE ID = :(ProductID)
```

- **Convertir valores en números enteros para cálculos.** Cuando quiera ejecutar algún cálculo numérico con las variables, asegúrese de convertir el valor de la variable en un número entero. Definir la variable como numérica únicamente limita los caracteres aceptados para esa variable, no cambia el tipo de variable. NiceLabel Automation trata todas las variables de tipo cadena. En VBScript, usará la función `CInt ( )`.
- **Configurar valores predeterminados / iniciales para secuencias de comandos.** Cuando use variables en la acción Ejecutar secuencia de comandos, asegúrese de que tengan un valor predeterminado asignado, o la comprobación de la secuencia puede fallar. Puede definir valores predeterminados en las propiedades de las variables, o dentro de la secuencia (y eliminarlos después de probar la secuencia).

## 9.20. Modo de rastreo

En forma predeterminada, NiceLabel Automation registra los eventos en la base de datos de registro. Esto incluye recopilar información de alto nivel, por ejemplo:

- registro de ejecución de acciones
- registro de ejecución de filtros
- registro de actualizaciones de estado de desencadenador

Para obtener más información, consulte la sección [Opciones de registro de eventos](#).

Sin embargo, el registro predeterminado no lleva un registro de las ejecuciones técnicas profundas. Cuando se necesita resolución de problemas en el nivel más bajo de la ejecución de códigos, debe activarse el modo de rastreo. En este modo NiceLabel Automation registra los detalles sobre todas las ejecuciones internas que se lleven a cabo durante el procesamiento de desencadenadores. El modo de rastreo solo debe activarse durante la resolución de problemas para recopilar registros y luego deshabilitarlos a fin de reanudar el normal funcionamiento.



### AVISO

El modo de rastreo retarda el procesamiento. Solo debe usarse cuando así lo indica el equipo de soporte técnico de NiceLabel.

## Habilitar el modo de rastreo

Para habilitar el modo de rastreo, realice lo siguiente:

1. Navegue a la carpeta NiceLabel Automation System.

```
%PROGRAMDATA%\NiceLabel\NiceLabel 10
```

2. Crear copia de seguridad del archivo `product.config`.
3. Abra el archivo `product.config` en el editor de texto. El archivo tiene una estructura XML.
4. Agregue el elemento `Common/Diagnostics/Tracing/Enabled` y asígnele el valor **Verdadero**. El archivo debe tener el siguiente contenido:

```
<?xml version="1.0" encoding="utf-8"?>
<configuration>
  <Common>
    <Diagnostics>
      <Tracing>
        <Enabled>True</Enabled>
        <Folder>c:\Troubleshooting\TracingLogs</Folder>
      </Tracing>
    </Diagnostics>
  </Common>
  ...
</configuration>
```

5. Después de guardar el archivo, el Servicio NiceLabel Automation aplica automáticamente la configuración.
6. En forma predeterminada, los archivos de rastreo (\*.LOG) aparecerán en la misma carpeta Sistema. Para sustituir la carpeta de registro, use el archivo `product.config`. Especifique la carpeta de registro personalizada en el elemento `Carpeta`. Este elemento es opcional.
7. Para confirmar la activación del modo de rastreo, abra Automation Manager. Con el modo de rastreo activado, mostrará el texto **Se activó el rastreo** en el panel de notificación sobre la lista de desencadenadores.

Después de reproducir el problema con el rastreo activado, desactive el rastreo en el archivo `product.config` o restaure su archivo `product.config` original.

```
<Tracing>
  <Enabled>False</Enabled>
```

## 9.21. Comprender la configuración de la impresora y DEVMODE



### NOTA

La estructura de datos DEVMODE es parte de la [estructura API de la impresora de GDI](#) en Windows. Esta sección incluye el contenido altamente técnico, relevantes sólo para los requisitos específicos.

Siempre que imprima una etiqueta en el software NiceLabel (o cualquier documento en aplicaciones de Windows), la aplicación de impresión leerá la configuración de la impresora según lo definido en el controlador de la impresora y la aplicará al trabajo de impresión. Se puede imprimir la misma etiqueta en diferentes impresoras con solo seleccionar otro controlador de impresora. Cada vez, la configuración de la nueva impresora se aplica a la etiqueta.

La impresión de un documento de texto en una u otra impresora láser generalmente produce el mismo resultado o uno muy similar. Imprimir etiquetas usando dos impresoras diferentes puede producir resultados muy inconsistentes. Para producir resultados comparables, el mismo archivo de etiquetas podría requerir ajustes adicionales del controlador de la impresora, tales como ajuste de desplazamientos, velocidad y temperatura de impresión. NiceLabel también aplica la configuración de la impresora en cada impresión. En forma predeterminada, se guarda la configuración de la impresora dentro del archivo de la etiqueta para la impresora seleccionada.

### ¿Qué es DEVMODE?

DEVMODE es una estructura de Windows que tiene la configuración de la impresora (información de entorno e inicialización sobre una impresora). Está compuesta por dos partes: pública y privada. La parte pública tiene datos comunes a todas las impresoras. La parte privada tiene datos específicos de cada impresora. La parte privada puede ser de longitud variable y tener toda la configuración específica del fabricante.

- **Parte pública:** Esta parte codifica la configuración general expuesta en el modelo del controlador de la impresora, como nombre de la impresora, versión del controlador, tamaño de papel, orientación, color, dúplex y otros. La parte pública es la misma para todos los controladores de impresoras y no admite trabajos específicos de impresoras de etiquetas (impresoras térmicas, impresoras de inyección de tinta industriales, máquinas de grabado por láser).
- **Parte privada:** Esta parte codifica la configuración no disponible de la parte pública. Los controladores de impresoras Loftware usan esta parte para almacenar los datos específicos del modelo de la impresora, como velocidad de impresión, temperatura, desplazamiento, modo de impresión, tipo de medios, sensores, cortadores, codificación de gráficos, soporte RFID y otros. La estructura de datos dentro de la parte privada de DEVMODE es una secuencia de datos binarios definidos por desarrolladores de controladores.

## Cambios en DEVMODE

La estructura de datos DEVMODE se almacena en el registro de Windows. Hay dos copias de la estructura: configuración de impresora predeterminada y configuración de impresora específica del usuario. Puede cambiar la estructura DEVMODE (configuración de la impresora) cambiando los parámetros del controlador de la impresora. Las primeras dos opciones están relacionadas con Windows, la tercera opción está disponible con el software NiceLabel.

- **Configuración de impresora predeterminada:** Esta configuración se define en **Propiedad es de impresora > pestaña Avanzado > Valores predeterminados de impresión**.
- **Configuración específica del usuario:** Se almacena en forma separada para cada usuario en la clave de registro HKEY\_CURRENT\_USER del usuario. En forma predeterminada, la configuración específica del usuario se hereda de la configuración predeterminada de la impresora. La configuración específica del usuario se define en **Propiedades de impresora>Preferencias**. Todas las modificaciones que se hagan aquí afectarán al usuario actual únicamente.
- **Configuración específica de la etiqueta:** El autor de la etiqueta que usa software NiceLabel puede insertar el DEVMODE en la etiqueta. De este modo, la configuración de la impresora es portátil. Cuando se copia la etiqueta a otra computadora, la configuración de la impresora se copia también. Para insertar la configuración de la impresora en la etiqueta, habilite la opción **Usar configuración de impresora personalizada guardada en la etiqueta** en *Propiedades de documento* en Designer Pro. Puede cambiar la configuración de las impresoras de etiquetas al seleccionar el panel *Impresora* en *Propiedades de documento*.

## Aplicar DEVMODE personalizada a la impresión

En NiceLabel Automation puede abrir un archivo de etiqueta y aplicar la estructura DEVMODE personalizada allí. Al imprimir la etiqueta, el diseño de la etiqueta se toma del archivo .NLBL y la estructura DEVMODE aplica el formato específico relacionado con la impresora. Esto permite tener una sola etiqueta maestra. La impresión será la misma independientemente de qué impresora utilice, ya que se usará la configuración óptima para esa impresora.

Para aplicar la estructura DEVMODE personalizada en la etiqueta, puede usar dos opciones:

1. Uso de la acción, más específicamente el parámetro **Configuración de la impresora**.
2. Archivo de comando JOB, más específicamente el comando **SETPRINTPARAM** con el parámetro **PRINTERSETTINGS**. Para obtener más información, consulte la sección [Usar comandos personalizados](#).

## 9.22. Usar la misma cuenta de usuario para configurar y ejecutar desencadenadores

El NiceLabel Automation servicio se ejecuta con las credenciales de la cuenta de usuario configurada durante el servicio. No obstante, Automation Builder siempre se ejecuta con las credenciales del usuario



registrado actualmente. Las credenciales de la cuenta de servicio y de la cuenta registrada pueden ser diferentes.

A pesar de que podrá generar la vista previa del desencadenador en Automation Builder sin problemas, el Servicio puede arrojar un mensaje de error, provocado porque las credenciales no coinciden. Si bien el usuario registrado actualmente tiene permiso para acceder a carpetas e impresoras, la cuenta de usuario que usa el Servicio puede no tenerlo.

Puede probar la ejecución de los desencadenadores en Automation Builder usando las mismas credenciales que tiene el Servicio. Para hacerlo, ejecute Automation Builder con la misma cuenta de usuario que está definida para el Servicio.

Para ejecutar Automation Builder con una cuenta de usuario diferente, realice los siguiente:

1. Mantenga presionada la tecla **Shift** y haga **clic con el botón derecho** en el Automation Builder icono.
2. Seleccione **Ejecutar con un usuario diferente**.
3. Ingrese las credenciales del mismo usuario, que se utiliza en el Servicio NiceLabel Automation.
4. Haga clic en **Aceptar**.

Si quiere ejecutar frecuentemente Automation Builder con credenciales de otra cuenta de usuario, consulte la utilidad de la línea de comandos **RUNAS** provista por Windows. Use los conmutadores **/user** para especificar la cuenta de usuario, y **/savecred**. Este último le permite ingresar la contraseña una única vez, y luego se recuerda la próxima vez.